ENERGY EFFICIENT DATA AGGREGATION IN WIRELESS SENSOR NETWORKS

Synopsis of the PhD Thesis

Submitted by

Archana Bomnale
SAP ID: 70821018002

Department of Electronics and Telecommunication Engineering

Supervisor

Dr. Avinash More

submitted in partial fulfilment of the requirements of

DOCTOR OF PHILOSOPHY (Engineering)

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING
SVKM's NARSEE MONJEE INSTITUTE OF MANAGEMENT STUDIES
(Declared as Deemed-to-be University Under Section 3 of the UGC Act, 1956)
V. L. Mehta Road, Vile Parle (West)
MUMBAI - 400056

Jul 2025

ABSTRACT

Wireless Sensor Networks (WSNs) are often deployed in harsh and hostile environments, which can affect the lifespan and dependability of individual sensor nodes considerably at times, leading to node failure. These node failures result in dynamic topology changes within the network, possibly disrupting established routes while transmitting data. Consequently, route discovery and maintenance have to be carried out constantly to maintain consistent and reliable communication in the network. Additionally, the random deployment of sensor nodes generates redundant data packets, leading to network congestion, communication delays, and inefficient energy utilization. Transmitting raw data further amplifies these inefficiencies, saturating the network and delaying communication. The nodes near the sink carry more traffic and deplete their energy faster as compared to farther nodes, that cause energy imbalance, leading to the "energy-hole problem." To overcome these challenges, we have proposed two novel protocols: Node Utilization Index-based Data Routing and Aggregation (NUIDRA), and Data Aggregation with Redundancy Removal (DARR). These protocols enhance routing reliability, optimize energy usage, eliminate data redundancy, and extend network lifespan by leveraging the following key innovations:

The first contribution is the design of a novel approach for computing the shortest routing paths for data transmission by formulating the node-utilization index (UI) based on the node-bandwidth consumption. By computing the UI, hop counts, and data aggregation factor, NUIDRA protocol ensures optimal routing while minimizing aggregation latency, solving the energy-hole problem and balancing the load traffic.

The second contribution is the computation of node-usage count which serves as a basis for the selection of aggregator nodes. These nodes aggregate the data by applying spatial correlation using Pearson correlation co-efficient, eliminating the redundant data.

The third contribution is the identification and scheduling of redundant nodes. DARR protocol identifies redundant nodes at the network level by utilizing spatial distance metrics to deactivate them to conserve energy, transitioning into a low-power (SLEEP) state until reactivation is required. The reactivation is done depending on the number of alive nodes that is checked periodically at Node-alive check time.

The fourth contribution is to eliminate data redundancy at node level by exploiting temporal correlations in the data sensed by the active nodes using DARR protocol. Further the DARR protocol is extended to High redundancy removal using temporal(DARR-HRT) to eliminate more redundant data. This temporally aggregated data is then transmitted to the sink node using shortest routing path. The DARR is integrated with NUIDRA protocol for implementing shortest path to transmit data and perform spatial aggregation at the aggregator nodes.

The simulation results demonstrate that the proposed Node-utilization index-based routing outperforms the existing methods by increase in 50% throughput. Similarly the proposed Node-alive index driven redundancy elimination method outperforms the existing techniques by extending the network lifetime upto 58%.

Keywords: Data Aggregation, Energy Efficiency, Network Lifetime, Redundancy, Wireless Sensor Network.

Contents

1 Introduction **1**
1.1 Motivation and Scope . . . 1
1.2 Problem Statement . . . 2
1.3 Research Objectives . . . 2

2 Literature Review **3**
2.1 Recent Works . . . 3
2.2 Research Gaps . . . 5

3 Node Utilization Index-based Data Routing and Aggregation (NUIDRA) Protocol **6**
3.1 Design and Implementation . . . 6
3.1.1 Phase I - Finding shortest path . . . 6
3.1.1.1 Computation of Shortest Path . . . 6
3.1.1.2 Computation of Node Utilization Index (UI_i) . . . 8
3.1.2 Phase II - Data Aggregation . . . 13
3.1.2.1 Selection of Aggregator Node . . . 13
3.1.2.2 Data aggregation using spatial correlation . . . 14
3.2 Result Analysis . . . 15
3.2.1 Average Throughput . . . 15
3.2.2 Packet Delivery Ratio . . . 16
3.2.3 Routing Overhead . . . 16
3.2.4 Aggregation Latency . . . 17
3.2.5 Number of Alive Nodes . . . 18
3.2.6 Average Residual Energy . . . 19
3.3 Summary of NUIDRA Protocol . . . 20

4 Data Aggregation with Redundancy Removal (DARR) Protocol **21**
4.1 Design and Implementation . . . 21
4.1.1 Phase I - Identifying Redundant Nodes . . . 21
4.1.2 Phase II Data aggregation using temporal correlation . . . 24
4.1.3 Extension of DARR protocol Data Aggregation with Redundancy Removal using High Redundancy Temporal (DARR-HRT) . . . 25
4.2 Result Analysis . . . 26
4.2.1 Percentange of Redundant Nodes . . . 27
4.2.2 Effect of Redundancy Threshold on Network Lifespan . . . 27
4.2.3 Energy consumption at source nodes . . . 28
4.2.4 Reduction in datasets using DARR-HRT . . . 29
4.2.5 Reduction in datasets after aggregation . . . 29
4.2.6 Data measurements sent . . . 30
4.2.7 Network Energy consumption . . . 30
4.2.8 Network Lifetime . . . 30
4.2.9 Quality of Service parameters . . . 31
4.3 Summary of DARR protocol . . . 34
4.4 Application Scenario . . . 34

5 Conclusion and Future Scope **36**
5.1 Conclusion 36
5.2 Future Scope 37

References **39**

List of Figures

3.1 Broadcasting route request packets from source node 7
3.2 Step 1 - Initialization of UI for all nodes 10
3.3 Step 2 - UI updated for path from S_1 to sink node 10
3.4 Step 3 - UI updated for path from S_2 to sink node 11
3.5 Alternate path - UI updated for path from S_{15} to sink node 11
3.6 Selection of Aggregator Node - UI updated for path from S_6 to sink node 14
3.7 Number of Nodes Vs Average throughput 16
3.8 Number of Nodes Vs Packet Delivery Ratio (PDR) 17
3.9 Number of Nodes Vs Routing Overhead 17
3.10 Number of Nodes Vs Aggregation Latency 18
3.11 Number of Nodes Alive over Time 19
3.12 Number of Rounds vs Average Residual Energy 20

4.1 Working mechanism of proposed DARR protocol 22
4.2 Identification of redundant nodes 23
4.3 Data aggregation using temporal correlation 24
4.4 Data aggregation using DARR-HRT 26
4.5 Redundancy threshold Vs Redundant nodes 27
4.6 Redundancy threshold Vs Network lifespan (sec) 28
4.7 Energy consumption(J) at source nodes 28
4.8 Reduction in datasets(bits) using HRT 29
4.9 Data measurements Vs Reduction in Data sets(kb) 29
4.10 Spatial correlation threshold (δ) vs. Data measurements sent (kb) % 30
4.11 Network Energy consumption (J) 31
4.12 Network Lifetime(sec) 31
4.13 Effect of varying Node-density on Packet Delivery Ratio 32
4.14 Effect of varying Node-density on Throughput 33
4.15 Analysis of End-to-End Delay over Time 33
4.16 Analysis of Residual Energy 34

List of Tables

2.1 Limitations in data aggregation mechanisms in the existing work 4

3.1 Shortest path route from source node 1 to sink node 9 7
3.2 Shortest path route from Source nodes to Sink node 8
3.3 Experimental parameter settings 15
3.4 Evaluation parameters for varying Number of Nodes (Packet Size of 512 bytes) 18
3.5 Number of Alive nodes for varying Time in seconds (Packet Size of 512 bytes) 19
3.6 Average Residual Energy for varying Number of Rounds (Packet Size of 512 bytes) . . . 20

4.1 Determining the redundant nodes 23
4.2 Experimental parameter settings 27

Chapter 1

Introduction

In wireless sensor networks (WSNs), the sensor nodes are usually battery-operated and have limited energy sources[1] WSN is often deployed in dynamic environments where nodes can join or leave the network, or malfunction at any instant. In a large network, due to the random deployment of a large number of nodes, many redundant data packets are delivered, increasing network traffic and causing delays [2]. More frequent route discovery and maintenance may be required due to dynamic topology changes that may disrupt established routes[3]. To meet these expectations and adjust to changing network conditions, routing and aggregation strategies must be robust and adaptable [4].

1.1 Motivation and Scope

Data aggregation in wireless sensor networks (WSNs) is a key concept which aims to improve the network's efficiency, longevity, and data accuracy. The motivation for data aggregation in Wireless Sensor Networks (WSNs) stems largely from the need to increase energy efficiency. Since sensor nodes in WSNs are usually battery-operated and placed in inaccessible or remote areas, it might be challenging to repair or recharge their power supplies. Large amounts of raw data transmission can quickly deplete a sensor node's battery since data transmission uses more energy than local data processing. By enabling nodes to analyze data locally, data aggregation lessens the volume of data transmitted over the network. The longevity of sensor nodes is greatly increased by this energy conservation through reduced data transmission, which also prolongs the network's overall operating life.

Redundancies are common in sensor data, especially in settings where numerous nodes capture similar data. Redundancies can be eliminated with the aid of aggregation algorithms, resulting in more representative and significant data summaries or averages. In addition to saving energy, the network guarantees that the users receive high-quality data that is immediately applicable for analysis and action by transmitting only the processed and relevant information.

Another important driver of data aggregation in WSNs is scalability, particularly when networks expand to hundreds of nodes. In these large networks, aggregating data facilitates the management and processing of massive amounts of information. Aggregation helps to reduce the data so that it is possible to provide thorough summaries to a central node or sink rather than flooding the network with individual data transmissions from every node. As a result, WSNs can grow efficiently without experiencing unnecessarily high transmission expenses or network delay.

Finally, data aggregation aids in balancing energy consumption across the network. Since no single node is in charge of transmitting all the data, aggregating data at intermediate nodes more equitably distributes energy usage. Data aggregation helps keep some nodes from using up their power too soon by balancing the energy burden in this way, extending the overall network's lifespan. Performing aggregation requires local data processing, and efficient algorithms are required to balance between the computational load of aggregation and the energy saved from reduced data transmission. In summary, energy conservation, data relevance, enhancing scalability, and energy balance are all made possible by data aggregation in WSNs. These factors all help to make the network more resilient and efficient.

1.2 Problem Statement

The problem statement of the proposed research is, **"to develop an energy efficient data aggregation mechanism for wireless sensor networks".** This problem statement will aim at :

- Reducing data redundancy through node-level aggregation, leveraging spatial and temporal correlations.
- Mitigating the energy-hole problem by redistributing traffic from over-utilized nodes near the sink to less-utilized nodes.

Addressing this problem will hold considerable practical significance in boosting the energy of resource-constraint wireless sensor nodes.

1.3 Research Objectives

To address the above mentioned problem statement, we have set the following Research Objectives (RO's):

1. **RO1 : To identify redundant nodes based on spatial distance among them for node scheduling at the network level.**
   Identify adjacent nodes within the network using their communication range and spatial distance. Determine redundant nodes among these adjacent nodes by applying a spatial distance-based technique and a predefined redundancy threshold. Schedule the identified redundant nodes for data transmission to the sink node, leveraging the count of alive nodes to optimize resource utilization and extend the network's operational lifetime.
2. **RO2 : To compute the shortest and most reliable routing path based on utilization of nodes.**
   Compute shortest path using reactive routing, reflecting the transmission load of a node, ensuring less-utilized nodes are prioritized to balance energy consumption.
3. **RO3 : To remove redundant data using temporal correlation at node level.**
   Define correlation thresholds to determine when data points are sufficiently similar to be considered redundant. Identify patterns in data collected over time to detect redundancies and only transmit significant deviations from these trends reducing the communication overhead.
4. **RO4 : To aggregate data among the nodes during transmission to sink node.**
   Allocate the aggregator node depending on the number of adjacent nodes contributing the data. Detect the linear relationship between the aggregated datasets and remove redundancy at network level to minimize the network energy consumption.

Chapter 2

Literature Review

The sensor nodes in wireless sensor networks (WSNs) typically have limited energy sources and are battery-operated. Reducing energy- usage, latency, and bandwidth is necessary to extend the networks lifetime. WSN is often deployed in dynamic environments where nodes in the network can join, leave, or fail at any instant. Dynamic topology changes can potentially disrupt established routes, necessitating more frequent route discovery and maintenance. Since a huge number of nodes are randomly deployed, a lot of redundant data packets are sent, which increases network traffic and creates delays. Robust and adaptable routing and aggregation techniques are necessary to meet these demands and adapt to shifting network conditions.

2.1 Recent Works

Most sensors transmit the data via multiple paths to aggregate it and forward it to other nodes. The shortest path multihop (SHM) algorithm discusses the shortest pathfinding algorithm based on the Breadth-First search approach [5]. The reactive routing protocols are discussed in [6, 7, 8]. Significant routing overhead can be produced by Dynamic source routing (DSR) protocol[9], particularly in very dynamic situations. In the DSR routing protocol, routines for route maintenance and discovery can result in a lot of control message exchanges, which use energy and bandwidth. Compared with DSR, the Adhoc On-Demand Distance vector (AODV) provided less overhead route maintenance and discovery. [3, 10]. Modified dynamic source routing (MDSR) protocol [11] is used to reduce the Route Reply (RREP) packets, and the header size of the data packets is fixed to reduce the control overhead.Data aggregation combines the sensed data by employing aggregation functions such as Maximum (Max), Minimum (Min), Average (Avg), Sum, Difference (Diff), Count [12] or other math functions such as Match, correlation coefficient technique [13] is used to exploit temporal and spatial redundancy and generate a single packet which is forwarded to the next node. The "MATCH" function may not be efficient for removing redundancy in very large datasets due to its computational overhead. Reduction in data redundancy will lead to an increased network lifetime and reduced latency [14].

The integration of Wireless Sensor Networks (WSNs) with the Internet of Things (IoT) enhances both scalability and energy efficiency. Contemporary WSNs are engineered for low power consumption, enabling prolonged operation in remote or inaccessible locations. As IoT ecosystems grow, new sensor nodes can be easily added, accommodating large, dynamic environments like smart cities and industrial facilities [15]. This convergence also strengthens real-time monitoring and automated responses, allowing IoT platforms to generate alerts or activate devices based on sensor input, thereby creating a complete feedback loop from data collection to action.

Table 2.1 shows the limitations identified and the challenges addressed in data aggregation mechanisms in the existing literature.

Table 2.1: Limitations in data aggregation mechanisms in the existing work

Citation, year	Journal,publication	Challenges addressed	Algorithm	Network Approach	Evaluation parameters	Aggregation method	Routing method	Limitations
[1] 2023	Computer Networks, Elsevier	Routing, Aggregation	EESM	Distributed	Network lifetime, control overhead, packet delivery ratio	Heuristic search algorithm	Multipath routing	pathfinding process is affected by dynamic environment
[16] 2023	Microprocessor. Microsystem., Elsevier	Routing	EE Pagasis-based	Distributed	Energy, Alive nodes	multiple chain-based	Fixed-Path routing	With large size network it can be difficult to create more chains
[17] 2022	Int. Jr. of Information Technology, Springer	Routing	EBARP	Centralized	Energy, delay, bandwidth	Fuzzy logic	Lagrange relaxation	Lagrange relaxed solution deviates considerably from true optimal solution, leading to suboptimal routing paths in network
[18] 2022	Journal of Computer Networks and Communications, Hindawi	Routing,	AODVI	Reactive	energy,routing overhead, throughput	Hybrid	forwarding probability	Route request packet are randomly controlled so routes may be instable
[19] 2021	IEEE Access, IEEE	Routing, Aggregation	Q-DAEER	Centralized	Energy, queue and control message overhead, delay	Quality value(Q) Learning	Reinforcement learning	convergence within a practical timeframe is challenging
[20] 2020	Wireless Networks, Springer	Routing,	DEER	Reactive	Energy consumption, message flow, Travel distance		Greedy algorithm	Greedy approach may have increased hop counts and higher communication costs.
[13] 2022	Electronics, MDPI	Aggregation, Redundancy	STCDRR	Centralized	Time-complexity, energy, aggr.ratio	JACCARD function, Pearson co-efficent	-	Jaccard coefficient may not adequately account for variability due environmental factors or measurement errors
[2] 2021	Computer Communications Elsevier	Redundancy	EMA	Centralized	Accuracy	Exponential moving average	-	Without coordination among nodes, similar or correlated data may be transmitted multiple times by different nodes, leading to unnecessary energy consumption and bandwidth usage
[21] 2022	Microprocess. Microsyst. Elsevier	Aggregation, Scheduling	CPS	Distributed	Collision, delay	Dijkstras algorithm	Tree-based	May result in lack of load balancing
[22] 2021	IEEE Internet of Things Journal, IEEE	Aggregation, Scheduling	CORD	Distributed	Duty cycle, time	LDA tree using Dijkstra's algorithm	SPT	LDA algorithm can become increasingly complex with scalable network

2.2 Research Gaps

The prominent research gaps identified are as follows.

- Routing paths from the source nodes to the sink node are disrupted due to dynamic topology changes caused by changing environments, and node failures.
- There is unbalanced traffic load, as the nodes near the sink node deplete their energy faster compared to nodes away from the sink node.
- High data traffic near the sink node causes loss of data, and the retransmission of the data packets cause transmission overhead, and delay in the network.
- As the nodes are randomly deployed, many of the nodes transmit the redundant data if they are spatially close, leading to the excess energy consumption.
- Most algorithms use location-aware methods (such as GPS based positioning) to calculate the spatial distances between nodes. Despite their effectiveness, these strategies have serious disadvantages:

  i) High Energy Costs: GPS-enabled nodes quickly deplete battery life, reducing the lifespan of the network.

  ii) Enhanced Hardware Complexity: GPS devices increase hardware limitations and deployment costs.

  iii) Scalability Problems: Because coordinates change frequently in dense or dynamic networks, location-aware protocols have difficulty keeping up.

Chapter 3

Node Utilization Index-based Data Routing and Aggregation (NUIDRA) Protocol

Data aggregation slows the communication process and consumes many resources available to sensor nodes. So, for transmitting the aggregated data, most of the algorithms face challenge to identify the shortest and most reliable route. However, the nodes closer to the sink node are heavily utilized in multi-hop transmission and succumb to early death, which causes energy imbalance and an energy-hole problem in the network. To overcome this problem, we propose an innovative Node Utilization Index-based Data Routing and Aggregation (NUIDRA) protocol. The NUIDRA protocol is designed and implemented in two phases. The first phase is to determine the shortest path from the source node to the sink node based on the amount of bandwidth utilized by each sensor node, minimum hop count, node's residual energy, and data aggregation factor. In the second phase, the selected shortest path is used for data transmission and aggregation using the dynamic selection of the aggregator node. The choice of aggregator node is based on the node's Utilization Index (UI) and adjacent node count from which it receives the data.

3.1 Design and Implementation

The proposed NUIDRA protocol is designed and implemented in two phases and is shown in figure 3.1. The first phase is to determine the shortest path from the source node to the sink node within the network field. During the second phase, the selected path is used for data transmission and aggregation.

3.1.1 Phase I - Finding shortest path

Phase I of the NUIDRA protocol determines the shortest and most reliable path between the source node and the sink node. This path is based on the node's Utilization Index (UI) by choosing the most eligible nodes and ensuring the minimum hop count for data transmission on the selected path.

3.1.1.1 Computation of Shortest Path

After establishing the network, the sink node receives control packets through multiple paths from source nodes. The source node can be anywhere at random, unlike the leaf node in a tree. The shortest path starts from the source node and leads to the sink node. The shortest path discovery is initiated by broadcasting RREQ (Route Request) packets from the source node, as shown in Figure 3.1. When the sink node receives the RREQ packets, it sends replies by RREP (Route Reply) packets through the paths received from the source node. Each node marks its predecessor node as a *visited node* so that its predecessor is not revisited and stores the node count in the VisQ buffer as shown in Table 3.1. Initially, the hop count of the source node is considered as 0. The path through which the RREQ (Route Request) packet reaches the sink node at the earliest and the node whose predecessor

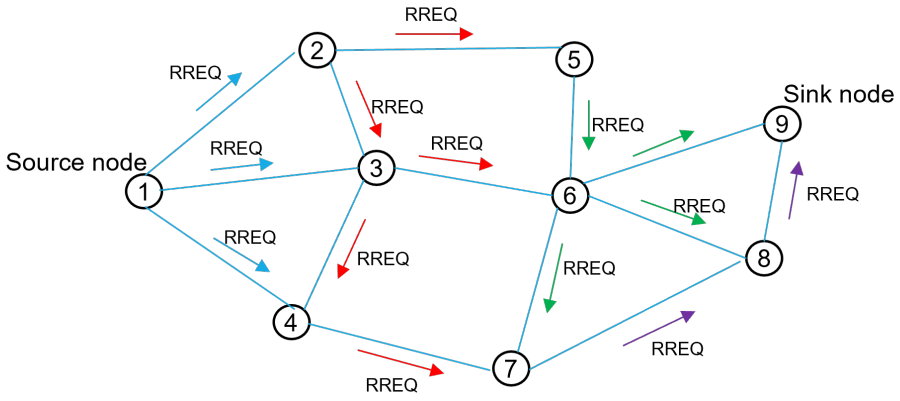

Figure 3.1: Broadcasting route request packets from source node

Table 3.1: Shortest path route from source node 1 to sink node 9

VisQ (visted nodes list)	Hop count	ShortPQ (list of paths)	AdjQ (Adjacent nodes list)				
1	0	1	2	3	4		
2	1	1-2	3	5			
3	1	1-3	2	4	6		
4	1	1-4	3	7			
5	2	1-2-5	2	6			
6	2	1-3-6	3	5	7	8	9
7	2	1-4-7	4	6	8		
8	3	1-3-6-8	6	7	9		
9	3	1-3-6-9	-	-	-	-	

∗ 1,2,3...9 indicates the node id except for the Hop count-column

value is null (hop count is 0) is considered the shortest path from the source node to the sink node. Here, the sink node counts the number of hops from the node whose packets are received at the earliest by sending an RREP message on that path and considers it as the shortest path for that source node. Figure 3.1 illustrates the broadcasting of route-request packets in the network from the source node in the shortest path-finding method.

The details of the parameters mentioned in Table 3.1 are as follows. The letter Q in each term denotes the list or array to store the integer value.

- VisQ - It is the visited node list to store the visited node-id from the source node.
- ShortPQ - It stores the list of paths. It shows the advancement of the shortest path from the source node to each visited node in VisQ.
- AdjQ - It stores the list of adjacent nodes to each node in VisQ.

As shown in Figure 3.1, node 1 is the source node, and node 9 is the sink node. The route from the source node to the sink node may take more than one path. Let us consider the nodes are randomly deployed in the given area, and the shortest path is to be identified from the source node 1 to the sink node 9. To determine the shortest route, the source node 1 creates the adjacent node's list for the first hop (1 hop count), as shown in Table 3.1. Similarly, the nodes visited in the second hop (2 hops) create the adjacent node's Table by adding the previously visited nodes, as presented in Table 3.1. For example, the source node 1 has three adjacent nodes (2,3,4) in the first hop (1 hop), as shown in blue colour in figure 3.1. So, for the first hop or one hop, Node 2 has received RREQ from Node 1, which forms a path as node 1 to node 2 and its adjacent nodes are node 3 and node 5 as shown in Table 3.1. When node 3 receives RREQ from node 1 and its adjacent nodes (2,4,6) the path is from node 1 to node 3. Further, for node 4, the path is from node 1 to node 4 with adjacent nodes (3,7) in the first hop or one hop, as shown in Table 3.1. Similarly, RREQ is broadcasted for two-hop, three-hop and four-hop count, as shown in Figure 3.1 with red, green and purple colours, respectively.

Finally, from the source node, when the last hop is to the sink node, the sink detects the shortest path for the received packets and backtraces all the nodes till it reaches the source node. Each node in

Table 3.2: Shortest path route from Source nodes to Sink node

S_i (Source node)	Path to sink node	Node ID								
		1	2	3	4	5	**6**	7	8	9
1	1-3-6-9	1		1			**1**			1
2	2-5-6-9		1			1	**1**			1
3	3-6-9			1			**1**			1
4	4-7-8-9				1			1	1	1
5	5-6-9					1	**1**			1
6	6-9						**1**			1
7	7-6-9						**1**	1		1
8	8-9								1	1
NUC_i(Node usage count)		1	1	2	1	2	**6**	2	2	8

the backward or reverse path will also update its Utilization Index (UI_i) value as explained in equation 3.1. The sink node finds the shortest path from the source node to the sink node based on the (UI_i) value for each node.

Table 3.1 shows the adjacent node list from source node 1 to the sink (destination) node. Each node stores the predecessor node ID, i.e., the node from where it receives the data. The source node also maintains the list of adjacent nodes.

Table 3.2 shows each source node's path to the sink node. Each node also stores its path to the sink node and the number of times it is utilized in the shortest path calculation of various nodes. The last row gives the NUC_i value for each node. For example, the NUC_i value for node 6 is 6, indicating the busiest node, as shown in Table 3.2.

The NUIDRA protocol uses the Node Utilization Index to determine the shortest path based on the data bits transmitted by the node and the aggregation factor. The evaluation of the Node Utilization Index is described below.

3.1.1.2 Computation of Node Utilization Index (UI_i)

The Node Utilization Index (UI_i) assigns the numerical weight to each node of a linked set of nodes. When the sink node finds the shortest path from the source node, the utilization index (UI_i) for each node in the selected path is updated. This process is repeated after finding the path from each source node to the sink node. The node Utilization Index (UI_i) is computed based on equation 3.1.

$$UI_i = 1 - [\frac{DSTpersecond * (1 - AF)}{DTC * LF_i}] \tag{3.1}$$

The terms used in equation 3.1 are defined below-

• **Data size for transmission (DST) -**

It is the number of data bits for transmission by the node.

• **Data transmission capacity of a node (DTC) -**

It is the total capacity of the node to transmit the maximum number of bits (number of data packets * packet size) per second.

• **Load-balancing Factor (LF_i) -**

LF_i is a node level constant and its value may vary from node to node. Default value of load-balancing factor is 1. Load-balancing factor is useful when data transmission capacity of node (DTC) need to be reduced to prolong the life of that node.

When the value of LF_i for i_{th} node is 1, full data transmission capacity (DTC) will be available for computing shortest path. When LF_i reduces (for ex. from 1 to 0.6) accordingly transmission capacity (DTC) will reduce to 60%. Thus 60% DTC will be available for further iterations of computing shortest path. This is specifically useful during network operation, when a node is consuming more energy in data transmission and dying fast. By reducing the LF_i factor, data transmission from that node can be reduced and diverted through other nodes.

• **Aggregation Factor (AF) -**

It is the ratio of the total number of data bits reduced by aggregation to the total number of data bits before aggregation. The value of AF depends on the number of redundant data bits removed at the aggregator node using spatial data correlation.

$$AF = \frac{\text{Number of data bits reduced by aggregation}}{\text{Number of data bits before aggregation}} \tag{3.2}$$

• **Node Usage Count (NUC_i) -** It is the number of times, the node is used in calculating the shortest paths in the network.

Illustration of UI Evaluation:

To illustrate the computation of shortest path, we have considered the below values for node utilization index for i^{th} node as per equation 3.1.
DST = 1024 bytes = 1024*8 = 8192 bits ;
DTC = 163840 bits/sec;
$LF_i = 1$
AF = 0

$$UI_i = 1 - [\frac{(DSTpersecond) * (1 - AF)}{DTC * LF_i}] \tag{3.3}$$

where i = 1 to n, n = number of sensor nodes.

The node6 is a part of shortest path of node1 as shown in Figure 3.3. UI_6 changes from 1 to 0.95 as mentioned in equation 3.4.

$$UI_6 = 1 - [\frac{8192 * (1 - 0)}{(163840 * 1)}] = 0.95 \tag{3.4}$$

Similarly, for computing shortest path for node2, second-time allocation for node6, its UI_6 is 0.90 as shown in equation 3.5.

$$UI_6 = 1 - [\frac{8192 * 2}{(163840 * 1)}] = 0.90 \tag{3.5}$$

Further, for computing shortest path for node3, node 6 is allocated third time, its UI_6 is 0.85 as shown in equation 3.6.

$$UI_6 = 1 - [\frac{8192 * 3}{(163840 * 1)}] = 0.85 \tag{3.6}$$

We have discussed the various steps to find the shortest path based on the node's Utilization Index (UI) from the source node to the sink node within the network as shown below.

Step 1: Initialization of Utilization Index

In the shortest reliable path data forwarding process, initially, each node's Utilization Index is considered as one $(UI - i) = 1)$, and the node usage count is set to zero, $(NUC_i = 0)$ as shown in Figure 3.2. Each node broadcasts a RREQ route request message to adjacent nodes. Based on the transmitted data bits per second, the node's Utilization Index (UI) is computed. Figure 3.2 shows the initial Utilization Index and node usage count for all the nodes in the network.

Step 2: Updated Utilization Index for the nodes in the shortest path from source node S_1 to sink node

The sink node receives the HELLO (RREQ) packets from all the nodes. The path through which the first HELLO packet is obtained from the source node is considered the shortest path for that source node. As shown in Figure 3.3, the sink node confirms the shortest path from source node 1 to sink and unicasts the RREP route reply packets to all the nodes in this path. At the same time, each node (6,3,1) in this path updates its UI value (0.95) as per equation 3.3.

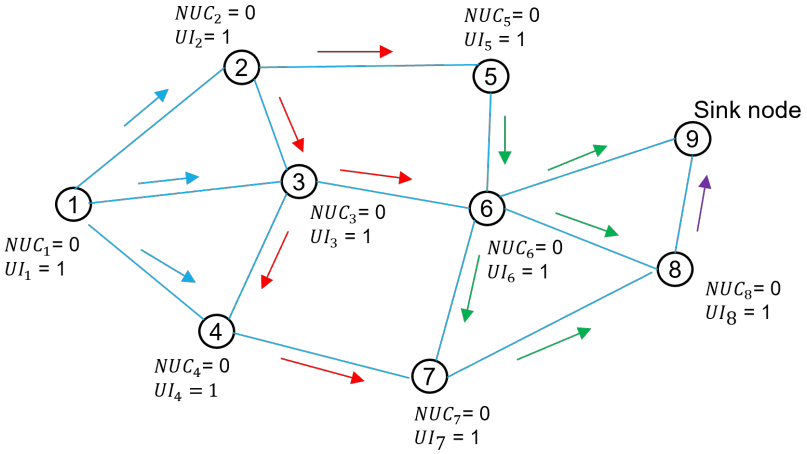

Figure 3.2: Step 1 - Initialization of UI for all nodes

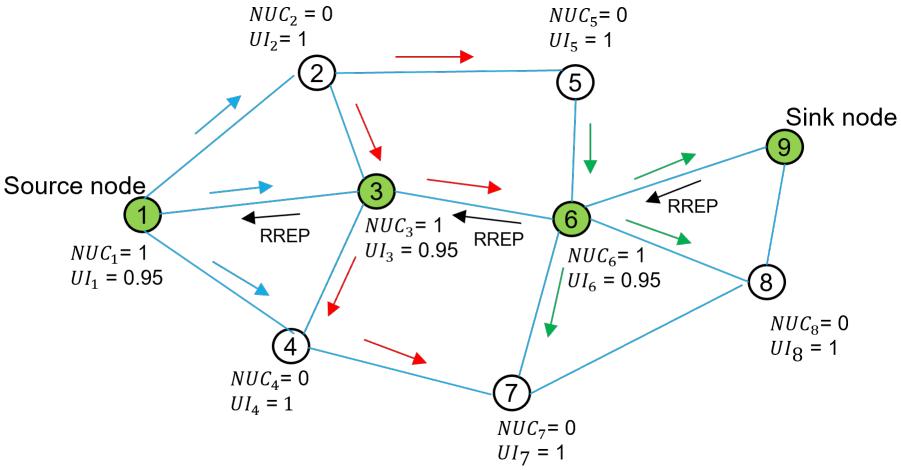

Figure 3.3: Step 2 - UI updated for path from S_1 to sink node

Step 3: Updated Utilization Index for the nodes in the shortest path from source node S_2 to sink node

The sink node confirms the shortest path from node 2 and unicasts the RREP to all the nodes (6,5,2) in the path. At the same time, each node in this path (6,5,2) updates its UI values (0.90, 0.95, 0.95) as per equation 3.3 as shown in Figure 3.4. The UI value of Node 6 is 0.90 because it is utilized twice for finding the shortest path. Node 6 was earlier used for shortest path finding from source node 1 to sink node as discussed in Figure 3.4.

The same procedure is repeated till the algorithm finds the shortest path for each node.

Alternate path (Re-routing the path) to node 6

As shown in Figure 3.5, the UI index of node 6 ($UI - 6$) has reached zero, so the packets do not take the route through node 6 to find the shortest path from sink node to source node 15(S_{15}). Thus, the sink node finds the alternate path (node 8,7,4,1,15) with UI_i values (0.95, 0.90, 0.80, 0.60, 0.95) respectively, making the transmission of data packets reliable.

Load Balancing :

In a wireless sensor network, the nodes near the sink are overutilized, and the nodes away from the sink are underutilized, causing unbalanced utilization of energy and data traffic. This may reduce the network lifetime. We have balanced the energy and traffic load in the network by computing the load-balancing factor (LF). The value of LF can range from 0 to 1. In the computation of the shortest path, initially, the load-balancing factor has a value of 1 for every node.

The node closest to the sink node has more significant traffic, which causes its UI to hit zero earlier

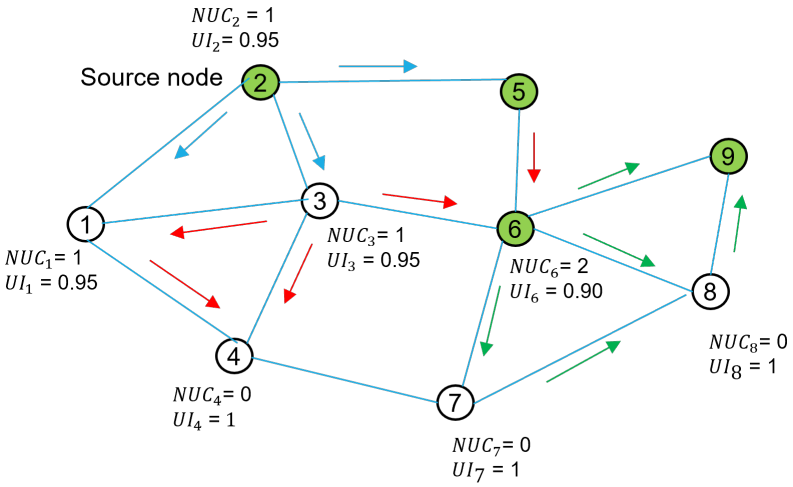

Figure 3.4: Step 3 - UI updated for path from S_2 to sink node

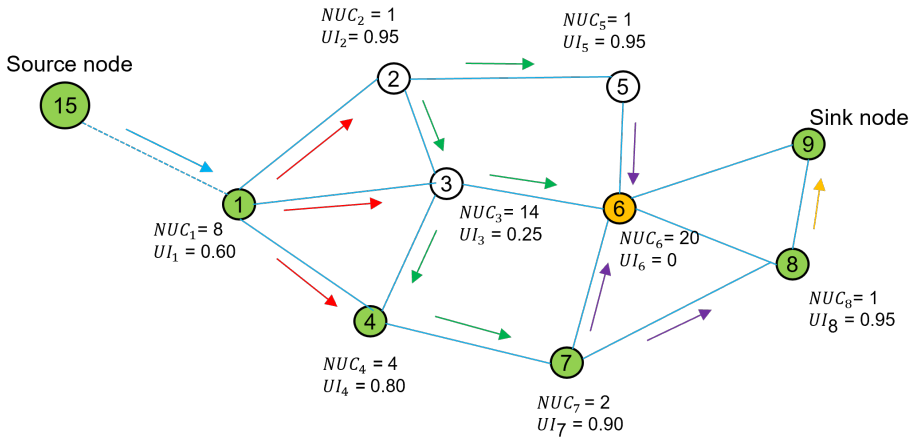

Figure 3.5: Alternate path - UI updated for path from S_{15} to sink node

than the other nodes. As shown in Figure 3.4, Node 6 may die prematurely since it is close to the sink and receives more data from nearby nodes than other nodes. The shortest path is recalculated once Node 6 dies. Node 1, Node 4, and Node 7 are underutilized, as seen in Figure 3.4, whereas Node 6 is overutilized. Balancing the energy and rerouting the data traffic leads to less energy consumption and increased network lifetime.

Recalculation of shortest path :
The frequency of calculation of shortest paths depends on the value of β. β is the ratio of the number of alive nodes to the total nodes in the previous shortest path calculation.

$$\beta = \frac{\text{Number of alive nodes}}{\text{Total nodes in previous shortest path}} \tag{3.7}$$

Let us consider there are 100 sensor nodes in the network while calculating the last shortest path. When 93 nodes are alive, and seven nodes are dead, the β value is 0.93. The Alive-node-threshold (τ)value is set to 0.95. Whenever β will be less than τ, the network will recalculate the shortest paths for all nodes.

The value of the Alive-node threshold can be increased or decreased depending on the network's need for reliability. Alive-node-threshold value is decided at the time of network initialization. If we want the network to be more reliable, we consider the maximum value of Alive-node-threshold and hence increase the frequency of shortest path calculation.

Algorithm 1 shows the process of creating the shortest reliable path from the source node to the sink node.

Algorithm 1: Finding shortest path for all nodes

Input - Source node (S_i), Sink node, Total number of nodes(n), Data transmitted by the node(DTS), Aggregation factor(AF), Min. number of nodes to start aggregation(N_{min}), load-balancing factor(LF_i).

Output - ShortPQ (shortest path list), Node Utilization Index for i$^{\text{th}}$ node(UI_i).

1:**Start**
2: Initialize Node Usage Count NUC_i to 0 for all nodes
3: Initialize Node Utilization Index UI_i to 1 for all nodes
4: **Repeat** for all nodes
5: **Call Function: Calculate shortest path for each source node**
6: **End loop**

Function: Calculate shortest path for each source node

1:**Start**
2: LinkedList Q, AdjQ, VisQ, ShortPQ
3: Integer i=0
4: Q.add(Si)
5: **Repeat** till Q is empty – Q contains all nodes, repeat till all nodes processed
6: node = Q.remove() – Get node from Q
7: VisQ.add(node) – As node visited, add to visited list
8: ShortPQ.add(node)
9: node.findAdjNodes(AdjNode) – Find adjacent nodes for a node
10: **Repeat** for i < AdjNodes – Copy Adjacent nodes to AdjQ
11: AdjQ.add(node.AdjNode)
12: ++i
13: **End loop**
14: **Repeat** till AdjQ is empty – Repeat till all AdjQ nodes are processed
15: **If** node is sink node
16: **Call Function: Calculate UI** – Calculate UI for each node in shortest path
17: Return ShortPQ – Return shortest path

18: Go to Stop
19: **End If**
20: **If** node is not in VisQ and UI > 0 – If node is not visited and UI index is not fully utilized
21: Q.add(node) – add node to list for further processing
22: **End If**
23: **End loop**
24: **End loop**
25: **Stop**

Function: Calculate UI
1: **Start**
2: **Repeat** for each node in the shortest path
3: ++ NUC_i
4: **If** $NUC_i < N_{min}$
5: AF = 0
6: **Else**
7: AF = α – set aggregation factor constant
8: **End if**
9: UI = 1 - [(DST* (1-AF))/(DTC*LF_i)] – Compute node utilization index
10: **End loop**
11: **Return** ShortPQ

3.1.2 Phase II - Data Aggregation

Each source node sends the data to the next eligible node. The node becomes an *Aggregator Node* when it receives data from adjacent nodes as per the set criteria to become aggregator node and forwards the aggregated data to the sink node through multi-hop transmission. NUIDRA protocol computes the similarity between data sets and removes the redundant data using the Pearson correlation coefficient [13] at the aggregator node . The protocol sets two variables for the aggregator nodes. The aggregation at the network level depends on the value of these two variables.

- (N_{min}) : Number of minimum nodes required to start aggregation.
- (T_{max}) : Maximum waiting time to start aggregation.

For example, the protocol set the value of N_{min} as 4 and T_{max} as 200 milliseconds with a network of nine nodes. Aggregation will start when either of the above conditions is met earlier. The waiting time to start aggregation depends upon the size of the data packet transmitted by the node. If there are only three adjacent nodes instead of 4, then the aggregator node will wait for time T_{max} to receive data from one more node, for which the aggregator node will be a part of its shortest path. After waiting for time T_{max}, the aggregator node will start aggregation with three adjacent nodes.

3.1.2.1 Selection of Aggregator Node

As shown in Figure 3.6, the sink node is at one hop distance from source node 6. The sink node confirms the shortest path from node 6, and its UI value is updated. Node Usage Count NUC_i for each node is also updated.

The updated UI value for node 6, while considering the path from node 4 is 0.80 according to equation 3.1. However, as per the assumption for the criteria of aggregator node, If $NUC_i \geq N_{min}$, then AF = 0.4. Thus, the NUC_6 value is equal to N_{min} value, i.e. 4. So, based on equation 3.5, the UI value for node 6 is 0.88, which indicates that more number of data bits can be transmitted through node 6.

The protocol can vary the value N_{min}. If it minimize the N_{min}, the data aggregation frequency will increase, consuming more energy. If it maximize the N_{min}, data aggregation frequency will decrease, but the aggregation delay will be greater. So the N_{min} and T_{max} values should be optimally selected. The aggregator node eliminates the redundant data collected from various nodes, thus achieving more

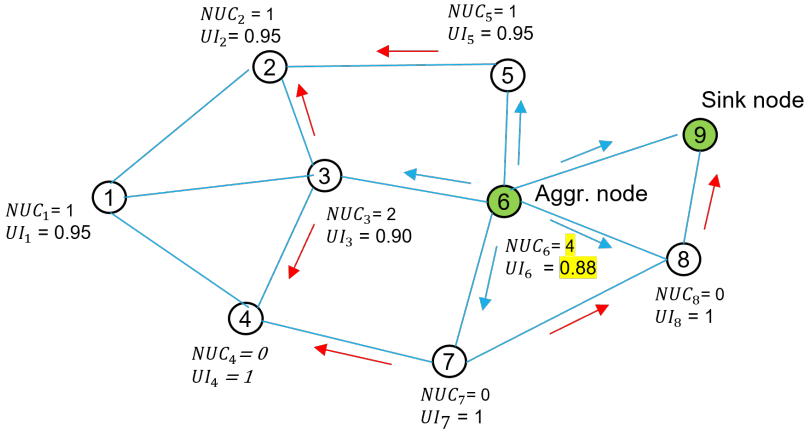

Figure 3.6: Selection of Aggregator Node - UI updated for path from S_6 to sink node

savings in the allocated node bandwidth.

3.1.2.2 Data aggregation using spatial correlation

The redundant data is eliminated at the aggregator node using Pearson correlation coefficient method [13]as given by equation 3.8, equation 3.9, and equation 3.10.

$$\rho_{X_i,Y_j} = \frac{\text{cov}(X_i, Y_j; wt)}{\sqrt{\text{cov}(X_i, X_i; wt)\,\text{cov}(Y_j, Y_j; wt)}} \tag{3.8}$$

$$cov(X_i, Y_j; wt) = \frac{\sum_{i=1}^{n} wt_i(X_i - m(X_i; wt))(Y_j - m(Y_j; wt))}{\sum_{i=1}^{n} wt_i} \tag{3.9}$$

$$m(X_i; wt) = \frac{\sum_{i=1}^{n} wt_i X_i}{\sum_{i=1}^{n} wt_i} \tag{3.10}$$

where ρ_{X_i,Y_j} is the Pearson correlation coefficient,
X_i, Y_j are the two data set variables,
cov(X_i, Y_j; wt) is the weighted covariance between X_i, Y_j,
m(X_i; wt) is the weighted mean of X_i.
By normalising the covariance of two variables, the statistical measure known as Pearson's Correlation Coefficient (ρ) approach quantifies the linear relationship between them. With findings ranging from -1 to 1, it assesses correlations between weighted sensor data streams:
-1: Perfect positive correlation
+1: Perfect negative correlation
0: No linear relationship
This technique is very helpful for examining redundancy in sensor networks, which fits well with the emphasis on protocol optimisation and data efficiency.

The selection of the aggregator node and transmitting the data to the sink node is explained using algorithm 2.

Algorithm 2: Data Aggregation

Input - Data from ith(D_i) node, Min. number of nodes(N_{min}), Max. waiting time to start aggregation(T_{max}), Node Usage Count(NUC_i), Waiting time interval(T_{wait})

Output - Node count for data reception(NCR), Aggregated data at next node(D_i)

1: **Start**
2: Initialize N_i, Linkedlist D_i
3: **If** $NUC_i \leq 1$ – If single node data then send without aggregation
4: Send D_i to N_{i+1} – Send data to next node

5: **Else If** $NUC_i < N_{min}$ – If nodes data less then N_{min} constant then send without aggregation. i.e. without eliminating redundant data
6: Receive D_i from nodes of multiple shortest paths
7: Send D_i to N_{i+1} – send data to next node
8: **Else If** $NUC_i \geq N_{min}$ – If nodes data greater then N_{min} constant then aggregate data, i.e. remove redundant data
9: Reset time t and node count NCR
10: **Step M**: Receive NCR as per shortest paths
11: **If** NCR $< N_{min}$ and $t < T_{max}$ – if condition is true wait for data to receive
12: t = t + T_{wait}
13: Wait for T_{wait} – Wait for specific time period
14: Go to step M – Go back to receive data step
15: **Endif**
16: Remove redundancy from data
17: Send D_i to N_{i+1}
18: **Endif**
19: **Stop**

3.2 Result Analysis

In this section, we have evaluated the performance of the proposed NUIDRA protocol using the NS2.35 simulator. We have run the simulation scenario ten times and presented the average results. It has been observed that the standard deviation values vary from ±2.5% for the evaluated parameters in the simulation. The simulation parameters considered for evaluating different protocols are as shown in Table 3.3.

Table 3.3: Experimental parameter settings

Parameters	Values
Simulation area	200*200 m^2
Number of sensor nodes	10 - 100
Number of sink node	1
Node-id size	1 byte
Mobility model	none
Control packet size	32 bytes
Data packet size	256, 512 bytes
Application Type	CBR
Initial energy	1 J
Mac protocol type	IEEE 802.11b
Bandwidth	1 Mbps
Frequency	2.472 GHz
Simulation time	600 s
Transmission Range	20 m

The performance of the proposed NUIDRA [23] protocol is evaluated against two protocols, QADA (Quality of service-Aware Data Aggregation)[24] and IoT-LEACH [25]. QoS parameters for WSN, such as Average Throughput, Packet Delivery Ratio, Routing Overhead, Aggregation Latency, Number of Alive nodes, and Average Residual Energy are analyzed.

3.2.1 Average Throughput

The efficiency of any routing protocol depends on the number of data packets sent to the sink node. Average throughput is the number of bits transmitted per unit time in the network. Figure 3.7 shows

the relation between the number of nodes and average throughput for various packet sizes. In NUIDRA, as the path from the source to the destination node is fixed for data transmission, only the nodes in the shortest path will receive the packets from the nodes on the path, and no other nodes can receive the data packets. Therefore, due to the single shortest path, traffic from different paths will not be accepted by these nodes, which leads to fewer packet drops. This increases the average throughput of the network as compared to other protocols. In QADA, the throughput is more than I-LEACH since it uses the hybrid model wherein cluster and tree are used to route the packets from the source node to the cluster head (CH) and CH to the sink node, respectively. For 100 deployed nodes, the average number of packets sent per second is 1110 kbps, 895 kbps, and 650 kbps for NUIDRA, QADA, and I-LEACH protocols, respectively, for a packet size of 512 bytes as discussed in Table 3.4. For a packet size of 256 bytes, the average throughput is less than a packet size of 512 bytes. A larger packet size allowed more data to be transmitted in each packet, increasing the average throughput.

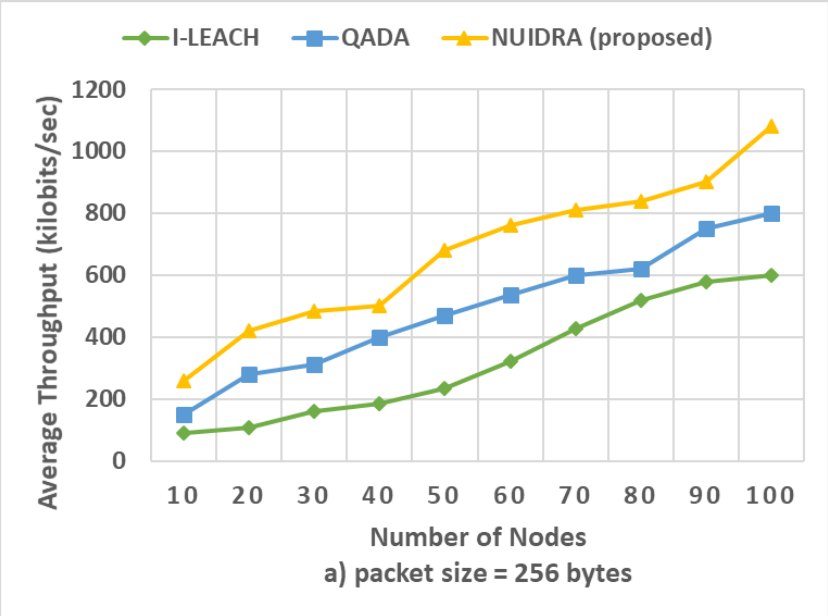

Figure 3.7: Number of Nodes Vs Average throughput

3.2.2 Packet Delivery Ratio

Packet delivery ratio (PDR) is the total number of data packets received at the destination to the total number of data packets sent from the source node. Figure 3.8 shows the PDR in NUIDRA is higher than other protocols, as the number of packets dropped is less. Nodes near the sink receive heavy traffic, which can determine the node usage and reroute the traffic accordingly in NUIDRA, which also avoids the energy hole problem. However, there is no provision to divert this traffic at the sink node in other protocols, which may cause energy hole problems and cascading effects. The packet delivery ratio is 90% for NUIDRA, 86% and 73% for QADA and I-LEACH protocols, respectively, with a packet size of 256 bytes for 100 deployed nodes. A higher packet delivery ratio indicates better network performance and more reliable communication, while a lower ratio suggests issues such as packet loss, congestion, or unreliable links within the network. With the change in packet size from 256 to 512 bytes, the PDR nearly remains constant for the NUIDRA protocol, whereas it decreases in the other compared protocols. Table 3.4 shows the values for PDR with packet size of 512 bytes for all the three compared protocols.

3.2.3 Routing Overhead

Figure 3.9 shows the number of nodes versus routing overhead. Routing overhead is the quantity of routing packets utilized for route maintenance and discovery. The routing packets for NUIDRA vary from 6 to 18, for the nodes varying from 10 to 100, whereas it varies from 5 to 15 and 8 to 19 for

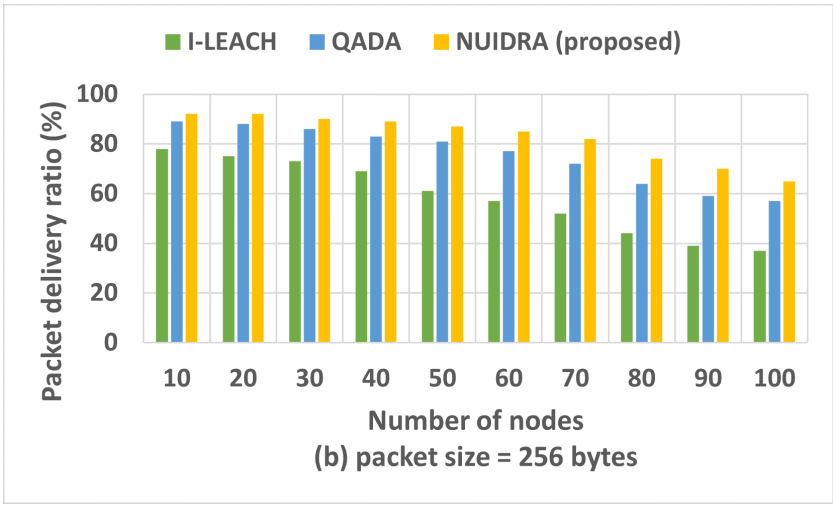

Figure 3.8: Number of Nodes Vs Packet Delivery Ratio (PDR)

I-LEACH and QADA protocol, respectively, for a packet size of 256 bytes. The routing packets for NUIDRA vary from 3 to 14, for the nodes varying from 10 to 100, whereas it varies from 2 to 12 and 5 to 17 for I-LEACH and QADA protocol, respectively, for a packet size of 512 bytes. Here, I-LEACH uses a clustered approach, and no shortest path is predetermined, so routing overhead is less. In the NUIDRA protocol, the routing tables must be updated and maintained by each node in the network, which requires memory and computing power. Thus, there is an increase in routing overhead as compared to I-LEACH. However, the QADA protocol has a higher computation overhead while constructing cluster-based trees for shortest-path routing than the NUIDRA protocol. The routing overhead for the protocols with a data packet size of 256 bytes is more than a packet size of 512 bytes. Data packets with 512 bytes result in fewer headers per unit of data, leading to a reduction in the overhead associated with packet processing and routing. The simulation results for routing overhead data packet size of 512 bytes is mentioned in table 3.4

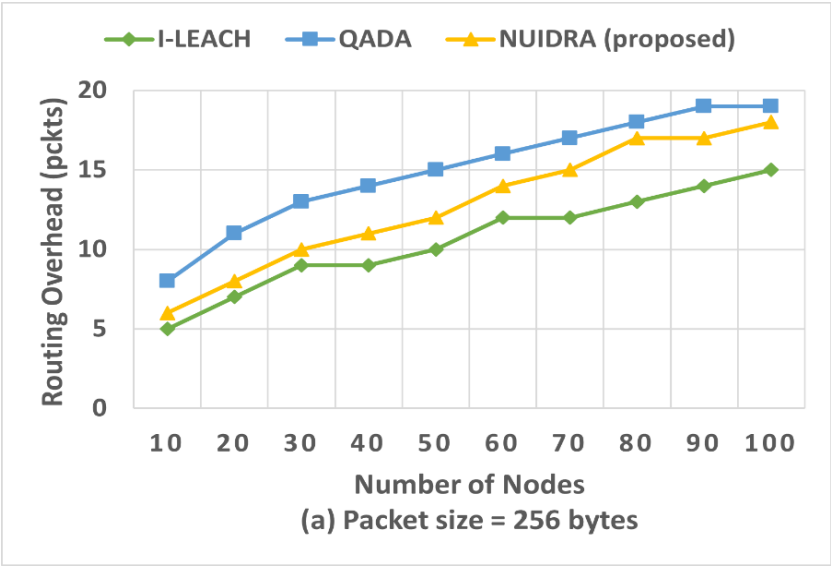

Figure 3.9: Number of Nodes Vs Routing Overhead

3.2.4 Aggregation Latency

Figure 3.10 shows the data aggregation latency as a function of the number of nodes. Aggregation latency is the time or delay before data from several sensor nodes is combined, summarised, or processed and then sent to a sink node. The average aggregation latency is lower with NUIDRA than with QADA and I-LEACH protocol because the data to be aggregated and transferred to the sink node travels via

Table 3.4: Evaluation parameters for varying Number of Nodes (Packet Size of 512 bytes)

Number of Nodes		10	20	30	40	50	60	70	80	90	100
Avgerage Through-put(kbits/sec)	I-LEACH	110	128	180	205	268	342	469	540	600	650
	QADA	190	320	350	420	510	580	630	700	810	895
	NUIDRA	280	450	520	580	710	790	790	870	930	1110
Packet Deliv-ery Ratio %	I-LEACH	72	65	62	58	50	46	41	39	37	35
	QADA	85	83	82	78	70	66	61	58	56	52
	NUIDRA	91	90	88	86	84	81	77	72	64	61
Routing Over-head (packets)	I-LEACH	2	4	6	6	7	9	9	10	11	12
	QADA	5	8	10	11	12	14	15	16	16	17
	NUIDRA	3	5	7	8	9	10	11	12	13	14
Aggregation Latency (mil-lisec)	I-LEACH	1200	1550	1641	1700	1800	1900	2200	2400	3300	3800
	QADA	650	880	1160	1350	1500	1700	1900	2050	3000	3400
	NUIDRA	400	570	684	717	825	1015	1250	1300	1450	1590

the shortest path, taking into account the node utilization index and aggregation factor. QADA reduces the redundant data, resulting in a lower average aggregation latency than I-LEACH. The NUIDRA protocol eliminates redundant data at the aggregator nodes and maintains approximately constant aggregation latency regardless of packet size. This is because once the shortest path is established, it remains unchanged for as long as a specific proportion of nodes are still alive, as noted in Section 3. Table 3.4 shows the simulation results for aggregation latency with a packet size of 512 bytes for all the three compared protocols.

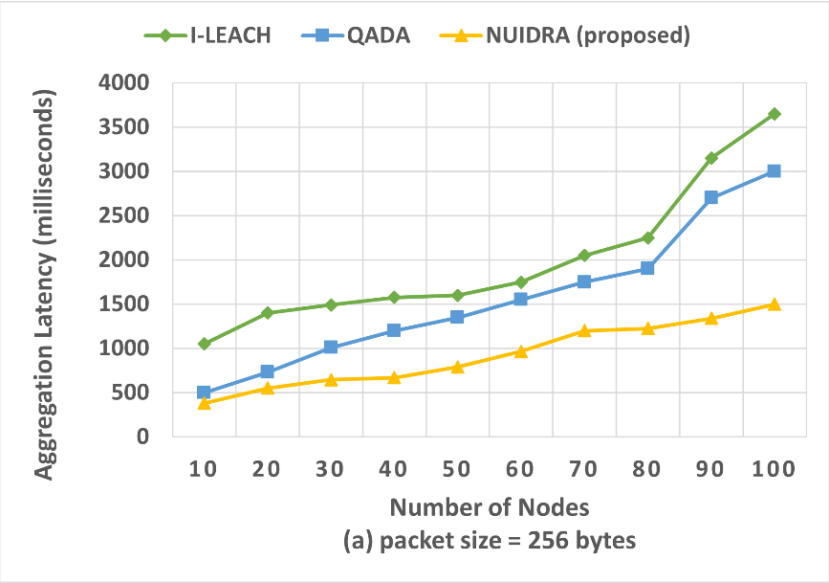

Figure 3.10: Number of Nodes Vs Aggregation Latency

3.2.5 Number of Alive Nodes

Transmitting larger packets can contribute to network congestion, especially in scenarios with a large number of nodes. Network traffic congestion may increase packet delays, potentially reducing the number of alive nodes. It can be seen from Figure 3.11 that the number of alive nodes decreases gradually in NUIDRA, compared to other protocols. There are fewer collisions in NUIDRA because

Table 3.5: Number of Alive nodes for varying Time in seconds (Packet Size of 512 bytes)

Normalized Time (sec)		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Number of Alive Nodes	I-LEACH	100	97	88	73	63	57	52	42	36	10
	QADA	100	98	91	80	71	68	61	55	47	18
	NUIDRA	100	100	95	91	86	81	74	61	51	26

data transmission uses a node utilization index-based shortest path. This prevents the retransmission of data packets, conserving energy, and hence, the number of nodes are alive for a longer period. In QADA, the number of alive nodes is higher compared to I-LEACH since it uses both the approaches cluster-based and tree-based routing that balances the energy load among nodes, whereas, in I-LEACH, CH directly communicates with the sink node in a single hop, which consumes sufficient network energy and drains the network early.

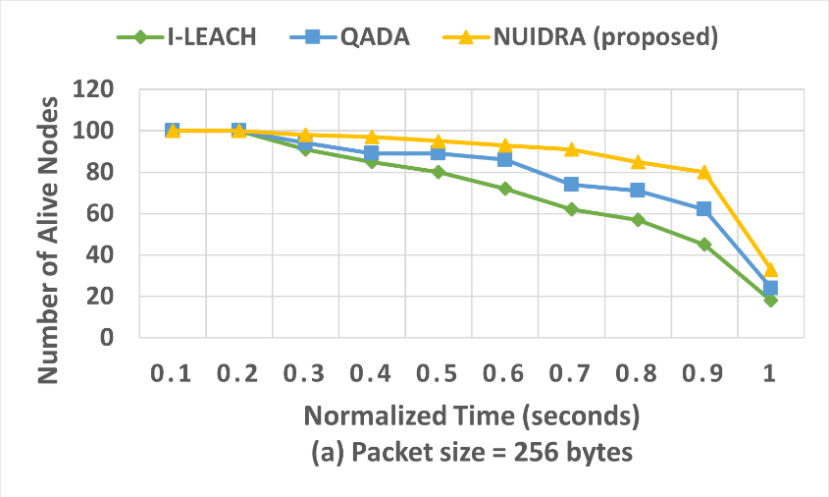

Figure 3.11: Number of Nodes Alive over Time

The simulated result for the number of alive nodes with compared protocols for data packet size of 512 bytes is indicated in Table 3.5.

3.2.6 Average Residual Energy

Average residual energy is determined by calculating the average of residual energy of all the sensor nodes in the network after any specific rounds of communication. Figure 3.12 depicts the graph of average residual energy versus number of rounds for packet size of 256 bytes. With the increase in number of rounds, the sensor nodes deplete their energy, so the average residual energy decreases, and the nodes die eventually. In comparison to QADA and I-LEACH, the NUIDRA takes the shortest path based on UI, which reduces the number of control packets, and hops in the multipath. When electing CH, QADA considers high energy parameters, and CH is used to establish the tree structure towards the sink node. As redundant data is removed during transmission by the aggregator node in NUIDRA and by the CH in QADA, more residual energy is still available in these protocols than in I-LEACH. In I-LEACH, the CHs are rotated in each round unless the CH's energy is above a certain threshold. As a result, every time the CH is changed, a lot of network energy is used. Additionally, in the I-LEACH protocol, the CHs consume more energy to send aggregated data in a one-hop transmission to the sink node. Residual energy is high after transmission of small packet size as compared to large packet size. The simulated result is indicated in Table 3.6 for a data packet size of 512 bytes.

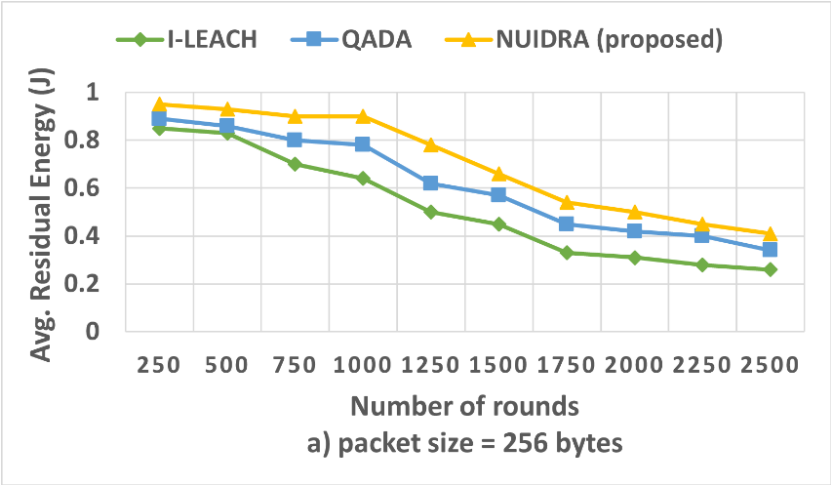

Figure 3.12: Number of Rounds vs Average Residual Energy

Table 3.6: Average Residual Energy for varying Number of Rounds (Packet Size of 512 bytes)

Number of Rounds		250	500	750	1000	1250	1500	1750	2000	2250	2500
Average Residual Energy (Joules)	I-LEACH	0.84	0.83	0.6	0.54	0.42	0.34	0.3	0.27	0.15	0.09
	QADA	0.86	0.86	0.7	0.64	0.57	0.47	0.38	0.31	0.22	0.11
	NUIDRA	0.91	0.88	0.81	0.72	0.64	0.52	0.47	0.45	0.4	0.22

3.3 Summary of NUIDRA Protocol

The NUIDRA protocol is designed to ensure energy-efficient, reliable data routing and aggregation in wireless sensor networks by incorporating multiple optimization factors, including the Utilization Index (UI) of sensor nodes, minimum hop count, aggregation factor (AF), and load-balancing factor (LF). The LF, derived from a node's current energy consumption, is crucial for addressing the energy hole problem near the sink node. The protocol selects aggregator nodes based on AF, ensuring redundant data is removed before transmission, thereby reducing data transmissions, packet sizes, routing overhead, and overall network energy consumption. To meet application-specific requirements, the T-wait time for aggregation is dynamically adjusted, balancing network latency. Reliability is further enhanced by setting the Alive-node-threshold (τ) to a high value, effectively redistributing traffic from heavily utilized nodes (identified by their UI values) to less-used nodes, thereby balancing energy consumption and extending network lifespan. Extensive simulations using NS2.35 show that for a data packet size of 512 bytes, NUIDRA achieves a 41% increase in average throughput, a 43% improvement in packet delivery ratio, and a 139% reduction in latency compared to I-LEACH, and a 19% increase in throughput, a 15% improvement in packet delivery ratio, and a 113% reduction in latency compared to QADA, highlighting its effectiveness in enhancing network performance.

Chapter 4

Data Aggregation with Redundancy Removal (DARR) Protocol

In most existing algorithms the data from all the nodes in a cluster is sent to the cluster head (CH) or aggregator node. There is a significant limitation in the existing algorithms, as all the nodes participate in data transmission to the CH or aggregator node. Over time, nodes may exhaust their energy due to continuous sensing, aggregating, and transmitting of data, eventually leading to their failure. These dead nodes can disrupt data transmission by hindering communication with intermediate nodes. This results in large consumption of network energy. To overcome these challenges, we have proposed the "Data Aggregation with Redundancy Removal" (DARR) protocol.

4.1 Design and Implementation

We present a two-phase design and implementation approach for the DARR protocol, illustrated in Figure 4.1.

In the first phase, adjacent and redundant nodes are identified based on spatial distance, and redundant nodes are set to Sleep state using our proposed DARR protocol [26]. These sleeping nodes do not participate in data sensing or transmission. Among active nodes, shortest path is computed using our proposed NUIDRA [23] protocol.

The second phase removes redundant data from source nodes through temporal correlation. The aggregated data is then transmitted to the next active or intermediate node, then to the aggregator node and finally to the sink node. The aggregator node removes the redundant data using spatial correlation [12]. The transmission of data from source nodes to the sink node takes the shortest route as computed in our second proposed NUIDRA protocol [23]. Further details on phases I and II are provided in the following subsection.

4.1.1 Phase I - Identifying Redundant Nodes

The sink node identifies adjacent nodes for each node based on the spatial distance (d_{im}) between them. This distance is measured using the Received Signal Strength Indicator (RSSI) readings [27]. Nodes are considered adjacent when their spatial distance is less than or equal to the communication range (R_c), as specified in Equation 4.1.

$$Ad_n = \left\{ \begin{array}{ll} 1 & \text{if } d_{im} \leq R_c \\ \\ 0 & \text{Otherwise} \end{array} \right\} \tag{4.1}$$

Here, i represents the current node, and m is an adjacent node. According to Equation 4.1, the adjacent-node flag is set to 1 when the distance d_{im} is less than or equal to R_c; otherwise, it is set to 0.

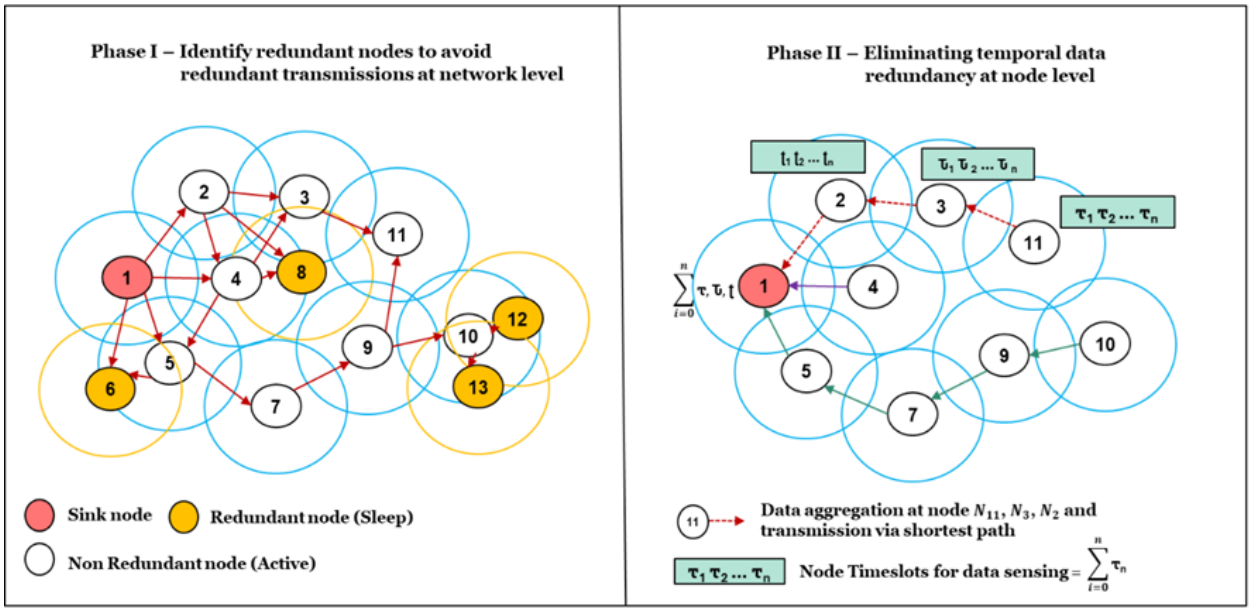
Phase I – Identify redundant nodes to avoid redundant transmissions at network level
Phase II – Eliminating temporal data redundancy at node level
Sink node
Redundant node (Sleep)
Non Redundant node (Active)
Data aggregation at node N_{11}, N_3, N_2 and transmission via shortest path
Node Timeslots for data sensing

Figure 4.1: Working mechanism of proposed DARR protocol

After identifying adjacent nodes, The proposed protocol identifies redundant nodes (R_n) using Equation 4.2.

$$R_n = \begin{cases} 1 & \text{if } (x * R_c)/d_{im} \geq 1 \\ 0 & \text{Otherwise} \end{cases} \tag{4.2}$$

Here, x represents the redundancy threshold, ranging from 0.1 to 0.9. The threshold determines how much overlap is acceptable in node coverage, depending on the application's needs.

Distance estimation

The log-distance path loss model [28] for RSSI-based distance estimation is defined by Equation 4.3

$$\text{RSSI}(d) = \text{RSSI}(d_0) - 10n \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma \tag{4.3}$$

where:

- RSSI(d): Measured signal strength (dBm) at distance d
- RSSI(d_0): Reference signal strength (dBm) at known distance d_0 (typically 1 meter)
- n: Path loss exponent (environment-dependent, n range from 2 to 6)
- d: Distance to estimate (meters)
- d_0: Reference distance (meters)
- X_σ: Gaussian noise with zero mean (models shadowing effects) and standard deviation is 5dB.

For example, Figure 4.2 illustrates a network of twelve sensor nodes. The sink node calculates the spatial distance (d_{im}) among all nodes using received signal strength data. Setting the redundancy threshold (x) to 0.45 and the communication range (R_c) to 10 meters, the sink node identifies adjacent nodes using Equation 4.1. It then applies Equation 4.2 to these adjacent nodes to determine redundant nodes. Table 4.1 provides the detail computation process, showing how the redundancy is calculated and represented across the network.

Step I : The sink node (Node 1) identifies its adjacent nodes: Nodes 2, 4, 5, and 6, as shown in Table 4.1. Starting with Node 2, the sink (Node 1) calculates the distance between them as 8 meters.

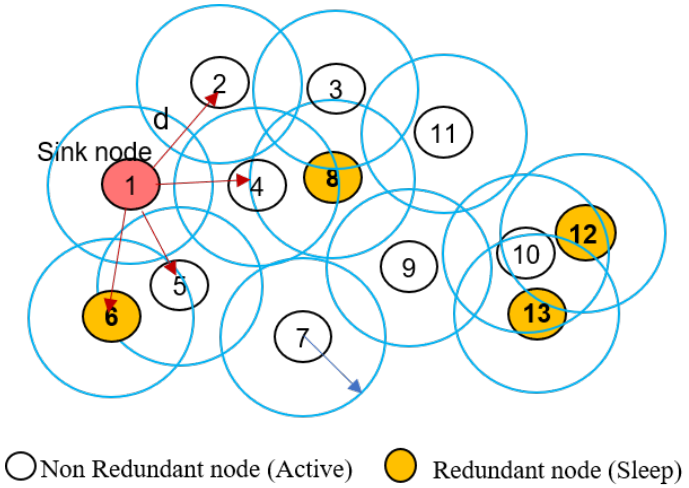

Figure 4.2: Identification of redundant nodes

Table 4.1: Determining the redundant nodes

Visited nodes	Sleep flag	Non-visited adjacent nodes	Redundant Node-count	Redundant Node-Id
1	0	2, 4, 5, 6	0	–
2	0	3, 4, 8	0	–
4	0	3, 5, 8	1	8
5	0	4, 6, 7	1	6
6	1	–	–	–
3	0	8, 11	0	–
8	1	–	–	–
7	0	9	–	–
11	0	9, 10	–	–
9	0	10, 13	–	–
10	0	12, 13	2	12, 13
12	1	–	–	–
13	1	–	–	–

Using Equation 4.2, the redundancy value (R_n) is found to be 0.56, which is less than 1. Therefore, the redundant flag is set to 0, indicating that sink (Node 1) and Node 2 are not redundant. The sink (Node 1) repeats this process for Nodes 4, 5, and 6, confirming none of them are redundant with sink (Node 1).

Step II : Node 2's adjacent nodes are Nodes 3, 4, and 8. The sink node calculates the distance between Node 2 and Node 4 as 6 meters. Applying Equation 4.2, the R_n value is 0.75, which remains below 1. Thus, Nodes 2 and 4 are not redundant. Similarly, the sink checks the redundancy for Node 3 and Node 8, confirming they are also not redundant with Node 2.

Step III : In Figure 4.2, Node 4 has adjacent nodes: Nodes 3, 8, and 5. The sink (node 1) calculates the distance between Node 4 and Node 8 as 4.5 meters. Using Equation 4.2, R_n equals 1, indicating that Node 4 and Node 8 are redundant. The sink updates the redundant flag accordingly and proceeds with the remaining adjacent nodes.

Step IV : Node 5s adjacent nodes are Nodes 4, 6, and 7. The distance between Node 5 and Node 6 is 4.5 meters. Applying Equation 4.2, is 1, so Nodes 5 and 6 are marked redundant. The sink performs similar checks for the remaining adjacent nodes to complete the analysis.

The sink node continues this iterative process (step I to step IV) to evaluate all adjacent nodes and identify redundant nodes. The results are maintained in a table at the sink node that records adjacent and redundant nodes, with redundant nodes highlighted (yellow) as shown in Figure 4.2. Table 4.1 lists specific nodes identified as redundant: Node 6 (with respect to Node 5), Node 8 (with

respect to Node 4), Node 12 (with respect to Node 10), and Node 13 (with respect to Node 10). Once redundant nodes are identified, they are switched to a Sleep state until the Node-Alive Check Time. Non-redundant nodes remain in the Active state during this period. Nodes alternate between Active and Sleep states depending on the number of operational nodes in the network. Active nodes handle data aggregation and transmit it to their adjacent nodes, and forward it to the sink node via the shortest path calculated by our NUIDRA protocol[23]. In the next section, we discuss the working of Phase II of our protocol.

4.1.2 Phase II Data aggregation using temporal correlation

Each source node collects data during a time interval (t) and performs aggregation. Each fixed time interval (t) is divided into j time slots (τ_j) during which the node senses and records data, as defined in Equation 4.4

$$t = \sum_{j=1}^{n} \tau_j \tag{4.4}$$

Here j is the time slot number within the interval (t), and n is the maximum number of time slots in the interval. As shown in Figure 4.3, sensor node 1 senses 10 events within a fixed time interval t, across time slots τ_1, τ_2, up to τ_{10}. The similarity measures threshold (ϵ) is set to 0.05. It measures the similarity between data values observed at a node over time. Data measurements in time slots τ_1, τ_2, and τ_3 are identified as redundant, and termed as Redundant Data Value. Similarly, time slots τ_4, τ_5, and τ_9, τ_{10} contain redundant data, while the remaining slots represent Non-Redundant Data Value.

The DARR protocol after aggregating data using temporal correlation, out of 10 sensed data measurements, only 6 data measurements are sent to the next intermediate node, as illustrated in Figure 4.3.

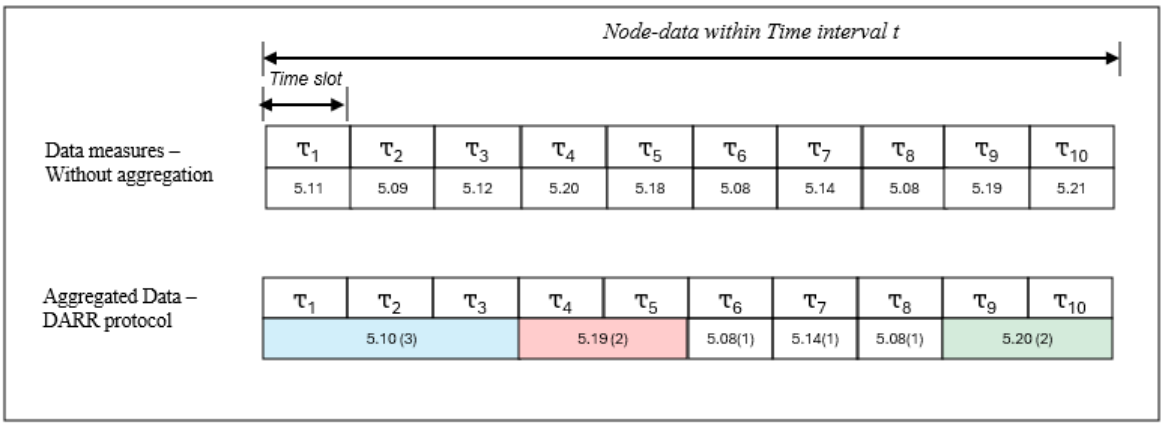
Node-data within Time interval t
Time slot
Data measures – Without aggregation
τ1 τ2 τ3 τ4 τ5 τ6 τ7 τ8 τ9 τ10
5.11 5.09 5.12 5.20 5.18 5.08 5.14 5.08 5.19 5.21
Aggregated Data – DARR protocol
τ1 τ2 τ3 τ4 τ5 τ6 τ7 τ8 τ9 τ10
5.10 (3)
5.19 (2)
5.08(1)
5.14(1)
5.08(1)
5.20 (2)

Figure 4.3: Data aggregation using temporal correlation

To identify similar data, the Cognate function is used, as shown in Equation 4.5

$$Cognate(S_{i,j}, S_{i,j+1}) = \left\{ \begin{array}{ll} 1 & \text{if } ||S_{i,j}\text{-}S_{i,j+1}|| \leq \epsilon \\ 0 & \text{otherwise} \end{array} \right\} \tag{4.5}$$

Here, S represents the sensor node, and (i, j) are integer variables representing the data of i^{th} node at j^{th} timeslot. The similarity measures threshold (ϵ), depends on the WSN application.

If data values are not redundant (Cognate function returns "0"), the mean is not updated, and the earlier mean and current data values are retained. After 10 time instants, the aggregated data is reduced from 10 initial data values to 6 non-redundant data values, with weights representing the number of occurrences of redundant data. This process illustrated in Figure 4.3 and explained in Algorithm 1 optimizes the data aggregation process by minimizing redundant transmissions and

improving network efficiency.

Algorithm1: Data Aggregation using temporal correlation

Input - DataA (data array contains j number of time slots with sensed data values),
TempA (array to keep data values to calculate mean).
Output - MeanA (array to store mean values),
WeightA (array to store weight for each mean value).

```
1: Start
2: Initialize array MeanA, WeightA, TempA and integers n=0, k=0, l=0
3: ϵ = 0.05 Constant for similarity measures threshold
4: TempAl = DataAn , MeanAk = DataAn , ++n Assign first data value
5: Repeat from 2nd value till jth value Last value in DataA
6:      If difference (DataAn, DataAn1) > ϵ If difference between current and
-                                                  previous value is more than ϵ
7:          l=0, ++k Initialize temp array l, increase array iterator
8:          Else
9:          ++l Increase iterator of temp array
10:          TempAl = DataAn
11:          MeanAk = Average (TempA0 . TempAl) Take mean of values and
-                                                          store in MeanA
12:          WeightAk = l +1  Store weight for above mean value
13:      Endif
14:      ++n
15: End loop
16: Return (MeanA array, WeightA array)  Returns aggregated data
17: Stop
```

For very high node density (more than 500 nodes), the DARR protocol causes higher computational overheads. DARR protocol does not capture the redundant data values if those are not consecutive in the given interval. To overcome these limitations, we have extended the DARR protocol with DARR-HRT mechanism.

4.1.3 Extension of DARR protocol Data Aggregation with Redundancy Removal using High Redundancy Temporal (DARR-HRT)

In the implementation of this modified protocol, DARR-HRT, phase I is the same as in the DARR protocol, and phase II is modified as discussed in the following.

The sensor node senses 10 data measurements. After aggregating the DARR protocol using temporal correlation, it provides 6 data measurements. By applying the modified DARR protocol with the High Redundancy Temporal (HRT) mechanism, this is further reduced to two transmitted data measurements, as illustrated in Figure 4.4

Figure 4.4 illustrates the reduction process, from ten raw data measurements to six aggregated data values using DARR protocol and then two aggregated data values using modified DARR protocol with high redundancy temporal (HRT) mechanism.

The identification and removal of redundant data from all the given data measurements using the DARR-HRT protocol is explained using Algorithm 2.

Algorithm2: Data Aggregation using DARR-HRT

Input - DataA (data array contains j number of time slots with sensed data values),
TempA (array to keep data values to calculate mean).
Output - MeanA (array to store mean values),

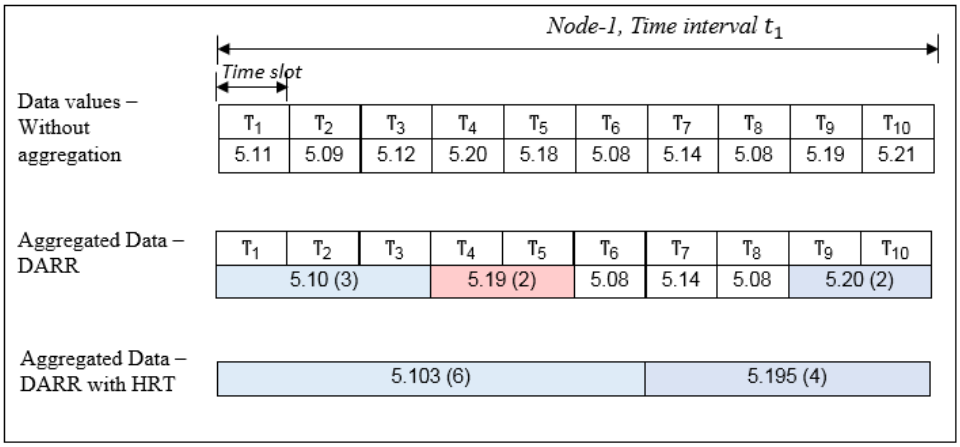
Node-1, Time interval t_1
Time slot
Data values – Without aggregation
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10
5.11 5.09 5.12 5.20 5.18 5.08 5.14 5.08 5.19 5.21
Aggregated Data – DARR
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10
5.10 (3) 5.19 (2) 5.08 5.14 5.08 5.20 (2)
Aggregated Data – DARR with HRT
5.103 (6) 5.195 (4)

Figure 4.4: Data aggregation using DARR-HRT

WeightA (array to store weight for each mean value).
1: **Start**
2: Initialize array MeanA, WeightA, TempA and integers n=0, k=0, l=0, m=0
3: $\epsilon = 0.05$ Constant for similarity measures threshold
4: TempAl = DataAn , MeanAk = DataAn , ++n Assign first data value
5: **Repeat** from 2nd value till jth value Last value in DataA
6: **Repeat** till m ¡= k Repeat for all mean values calculated
7: **If** difference (DataAn, MeanAm) $> \epsilon$ To check if difference is within - one of the mean values in the set
8: withinthreshold = Yes
9: Exit loop
10: **Else**
11: ++m
12: **Endif**
13: **End loop**
14: **If** withinthreshold = Yes
15: l = WeightAm
16: TempAml = DataAn
17: MeanAm = Average (TempAm0.. TempAml) - Take the mean values and - store in MeanA
18: ++WeightAm
19: **Else**
20: ++k
21: MeanAk = DataAn
22: WeightAk = 1
23: **Endif**
24: m=0, ++n, withinthreshold = No
25: **End loop**
26: **Return (MeanA array, WeightA array)** Returns aggregated data
27: **Stop**

4.2 Result Analysis

We have used the NS2 simulator [29] to assess the performance of the suggested DARR protocol. Table 4.2 displays the simulation parameters that are considered for evaluation. We have run the simulation scenario ten times and presented the average results. It has been observed that the standard deviation values vary from ±2.5% for the evaluated parameters in the simulation.

Table 4.2: Experimental parameter settings

Parameter	Numerical Value	Parameter	Numerical Value
Network area	200 * 200 sq. m	Mac protocol type	IEEE 802.11b
Number of sensor nodes	25 - 500	Bandwidth	1 Mbps
Number of sink node	1	Frequency	2.4 GHz
Node id size	1 byte	Simulation time	600 sec
Mobility model	None	Spatial correlation threshold (∂)	0.1 - 0.5
Control packet size	32 bytes	Transmission Range	20 m
Data packet size	256, 512 bytes	Redundancy threshold (x)	0.1 0.9
Application Type	CBR	Similarity measures threshold (ϵ)	0.01 0.1
Initial energy	1 J	Node-Alive threshold (Γ)	80 - 95

We have compared the performance of the proposed DARR [26] protocol with protocols, STCDRR [13], and DA-AFM [30]. The evaluation parameters for WSN, such as Percentage of Redundant nodes, Left-out nodes, Network Energy consumption, Reduction in data measurements, Data measurements transmitted, Network Lifespan, and Latency, are analyzed. During the simulation, the number of data measurements (M) is considered from 10 to 250.

4.2.1 Percentange of Redundant Nodes

The percentage of redundant nodes increases with the increase in redundancy threshold as shown in Figure 4.5, with the node density varying from 25, 50, 100, 150, and 200 nodes in the given network area.

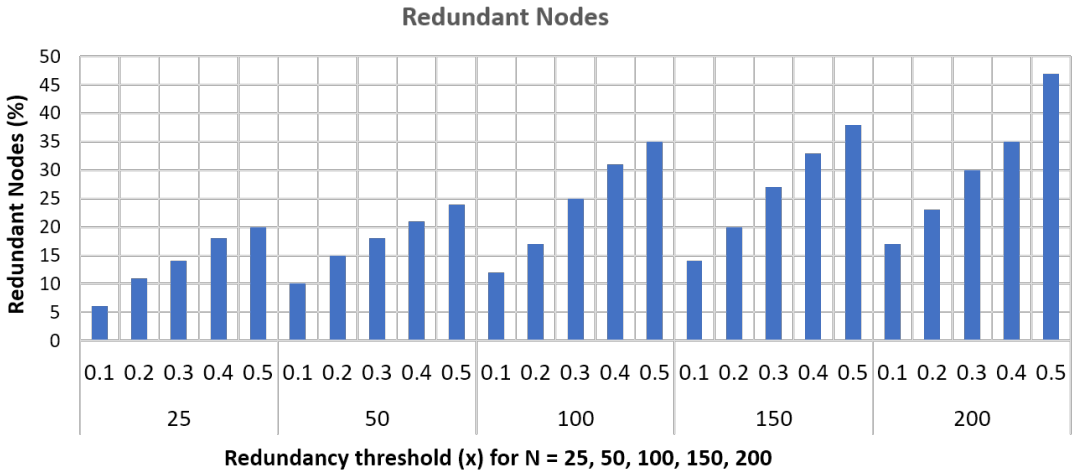

Figure 4.5: Redundancy threshold Vs Redundant nodes

Figure 4.5. shows that for 25 nodes, with redundancy threshold (x) as 0.1, the redundant nodes are 6% and with the value of x as 0.5, the redundant nodes are 20%. For 200 nodes, with redundancy threshold (x) of 0.1, the redundant nodes are 17%, and with x as 0.5, the redundant nodes are 47%.

4.2.2 Effect of Redundancy Threshold on Network Lifespan

Figure 4.6 shows the relationship between the redundancy threshold and network lifespan. If we increase the redundancy threshold, the more number of nodes becomes redundant. The DARR protocol sets all redundant nodes to Sleep state. The network lifetime is extended because more nodes in the

Sleep state prevent redundant data transmission from the sleeping nodes, which saves their energy. From Figure 7 4.6, we observe that with a network of 50 nodes, the normalized value of the average network lifespan is 0.19 and 0.71 when the redundancy threshold is set to 0.1 and 0.5, respectively. Similarly, with a network of 200 nodes, the normalized value of the average network lifespan is 0.3 and 0.95 when the redundancy threshold is set to 0.1 and 0.5, respectively.

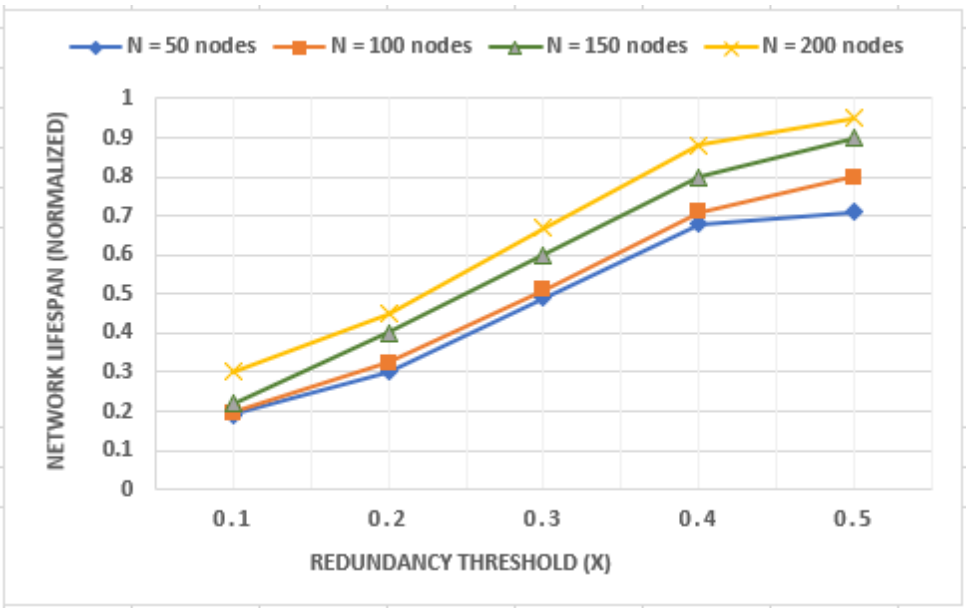

Figure 4.6: Redundancy threshold Vs Network lifespan (sec)

4.2.3 Energy consumption at source nodes

When the source nodes sense the data during the time interval (t), much of the sensed data is found to be redundant. The redundant data is not transmitted to the adjacent nodes, so the energy consumption of the source nodes is less. The value of the data sensing time interval (t) is defined by the application. The graph of the similarity measures threshold (ϵ) against the average normalized energy consumption at a randomly chosen source node-97 is displayed in Figure 4.7. We have considered twenty and fifty number of sensed data measurements (M) for simulation. With 50 data measurements, it was found that the normalized energy consumption reaches 100% when no aggregation (NA) is applied. In contrast, implementing temporal aggregation reduces energy usage: at a threshold value of 0.03, energy consumption drops to 90%, and at a threshold of 0.1, it further decreases to 40% compared to the scenario without aggregation.

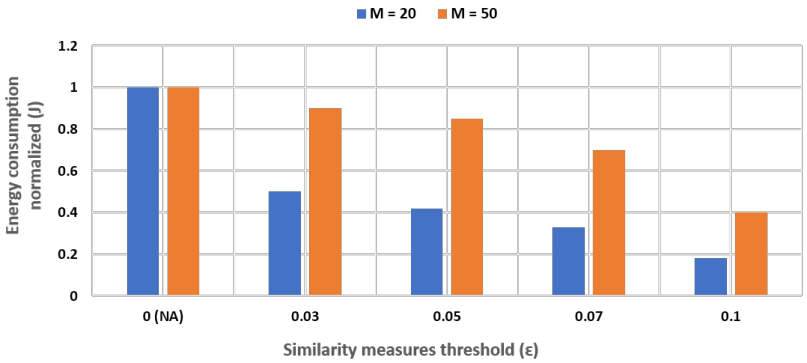

Figure 4.7: Energy consumption(J) at source nodes

4.2.4 Reduction in datasets using DARR-HRT

The percentage reduction in data measurements for DARR and DARR-HRT protocol is shown in Figure 4.8. When employing the DARR-HRT protocol, data measurements are reduced by 60 % for 250 data measurements, while the DARR protocol reduces data measurements by 38 % for a similarity measures threshold value of 0.05. However, there is a trade-off between the reduction in datasets and the accuracy of the retrieved data.

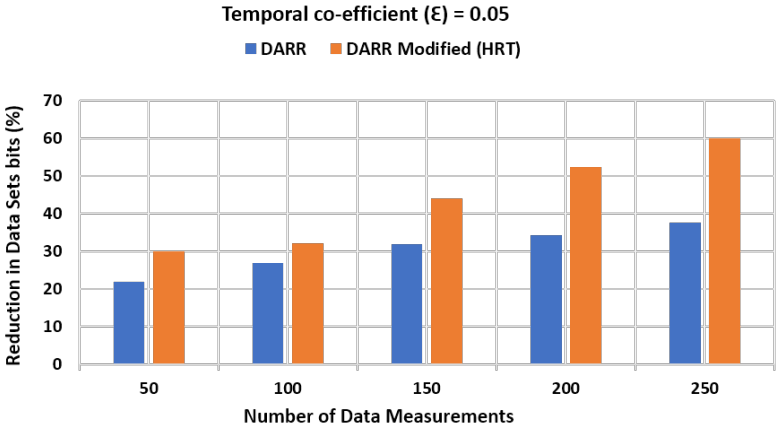

Figure 4.8: Reduction in datasets(bits) using HRT

4.2.5 Reduction in datasets after aggregation

The amount of redundant data removed at source nodes depends on the similarity measures threshold (ϵ). Redundancy in the data measurements increases as we increase the value of the similarity measures threshold. Figure 4.9 shows the percentage reduction in data measurements sent for varying values of several data measurements (M) from 10 to 50. The decrease in datasets for DARR, compared to STCDRR, and DA-AFM protocol is 12.5%, and 21%, respectively, for ϵ value set to 0.5 with 50 data measurements.

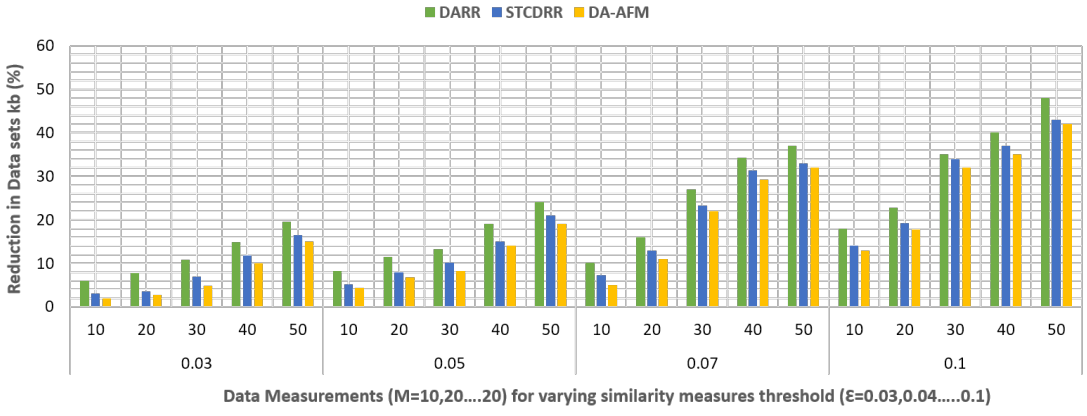

Figure 4.9: Data measurements Vs Reduction in Data sets(kb)

In the proposed temporal mechanism, the protocol takes the average of two successive data observations. Then, the average value is compared with the subsequent consecutive data measurement, so the difference between the mean and subsequent data measurement is minimized and thus we extract more redundant data, than the other two protocols.

4.2.6 Data measurements sent

Figure 4.10 shows the percentage of data measurements sent (kbits) for varying values of spatial correlation threshold (δ) for 200 nodes. Without data aggregation, the Active nodes send all the data measurements, including redundant data to the sink node so the network consumes more energy. The DARR protocol transmits comparatively fewer data measures than the other two protocols, that is 39% for DARR protocol, 52% and 64% as compared to without aggregation for STCDRR and DA-AFM method respectively. In the DARR protocol, the redundant nodes are identified in the network and put into a sleep state, and only the remaining active nodes send the data measurements. Since the redundant nodes are in a sleep state, they do not participate in sending the data, so only Active nodes send the data. This results in the transmission of very less data measurements to the sink node. In contrast, in the compared protocols, all the nodes send the data measurements to the cluster-head(CH), and then to the sink node.

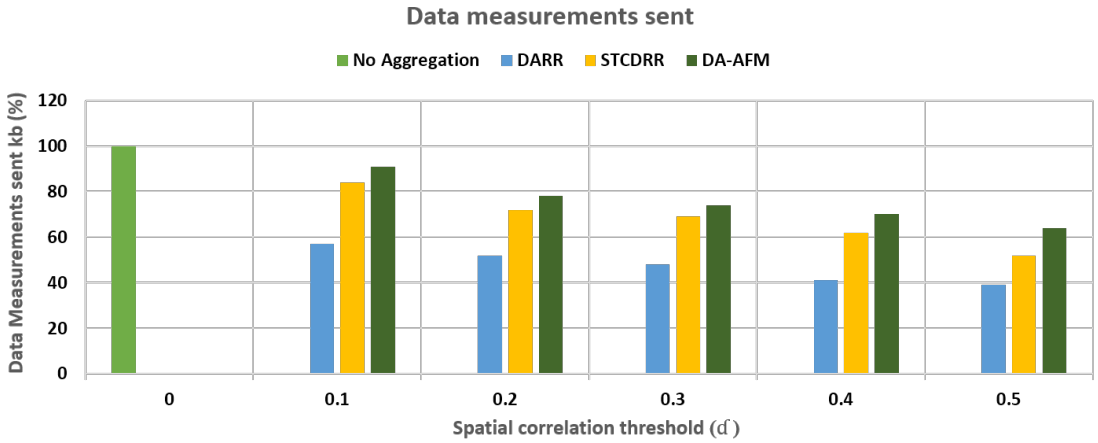

Figure 4.10: Spatial correlation threshold (δ) vs. Data measurements sent (kb) %

4.2.7 Network Energy consumption

When sensor nodes are densely populated, each node has numerous redundant neighbours due to the random distribution of sensor nodes. According to the DARR protocol, redundant nodes hold off on transmitting the data until the residual energy of the Active nodes drops below a certain threshold. Unlike the comparable protocols, this protocol saves network energy by setting all redundant nodes to the Sleep state. In the STCDRR and DA-AFM protocols, the data aggregation works at two levels, i.e., source node and cluster heads. In STCDRR, in the aggregation process, every node periodically reads the measurements of the data it has acquired and uses the Euclidean distance function to send its data set and weights to the CH. The periodic data transmission results in saving the network energy. The amount of network energy consumed increases with the number of data collection cycles or epochs.

Figure 4.11 shows the normalized network energy consumption for various data-aggregation rounds for the packet size of 512 bytes. In the 250th round, the network energy consumption(J) using the DARR-HRT and DARR protocol is 19% and 21% whereas, for the STCDRR, and DA-AFM technique, it is 54%, and 62%, respectively. As the number of data aggregation rounds further increases to 1000, normalized network energy consumption using DARR-HRT and DARR protocol is 34% and 44% whereas, for STCDRR, and DA-AFM technique, it is 71%, and 90% respectively.

4.2.8 Network Lifetime

Figure 4.12 shows the normalized network lifetime for the protocols discussed. The data aggregation technique of the DARR protocol reduces the number of data transmissions at three levels. First, redundant nodes are put into sleep mode, which prevents unnecessary data transmissions. Second,

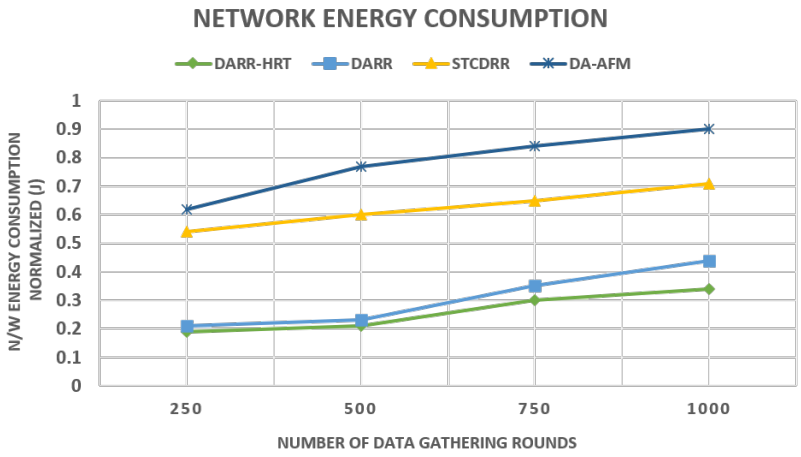

Figure 4.11: Network Energy consumption (J)

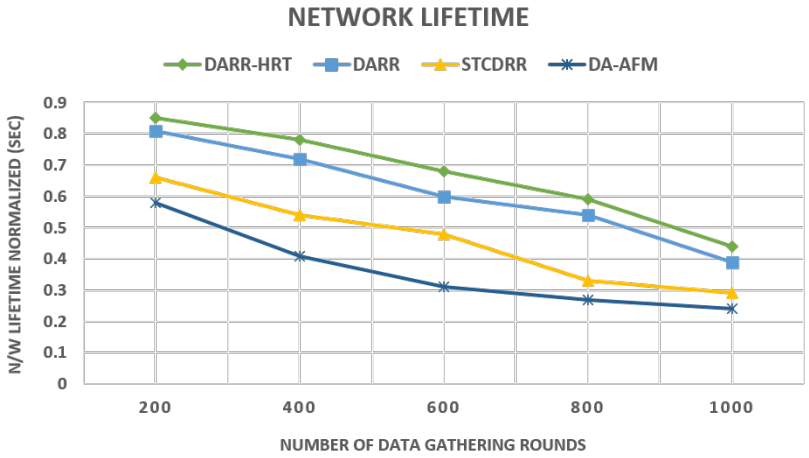

Figure 4.12: Network Lifetime(sec)

redundant data are removed through temporal correlation when source nodes aggregate data over a set period. The third, data are collected by the aggregator node using the shortest path as described in the proposed NUIDRA protocol, [23], and data aggregation is implemented at the aggregator node. This results in a decrease in network energy usage, extending the life of the network. The DARR protocol achieves a network lifetime that is 25%, 38% longer than the STCDRR, DA-AFM method and 12% shorter than DARR-HRT method over 1,000 data transmission rounds. The DA-AFM approach, which uses three-layer single-hop clusters in distributed sensor networks, has lower network lifetime due to increased data transmissions.

4.2.9 Quality of Service parameters

This section assesses the efficiency of MWA, QADA, and I-LEACH versus NUIDRA and DARR protocols.

Effect of varying Node-density on Packet Delivery Ratio

Figure 4.13 shows the number of nodes verses packet delivery ratio when 10,000 packets with a packet size of 512 bytes are transmitted.

It is observed that, DARR protocol uses temporal correlation to remove redundancy, improving PDR by 3-4 % and DARR-HRT improves PDR by 4-6 % by incorporating HRT mechanism over NUIDRA. MWA has a 68 % lower PDR than NUIDRA since node failures or congestion cannot be avoided via static cluster-based routes. In QADA, PDR is reduced by 10-20% due to cluster-tree routing's scalability issues, and IoT-LEACH provides 16-34% less PDR than NUIDRA since in dense

networks, it causes packet loss and collisions due to its reliance on fixed TDMA schedules and frequent cluster-head rotation. NUIDRA reduces transmissions and collisions by proactively eliminating upto 40% redundant data using Pearson correlation.

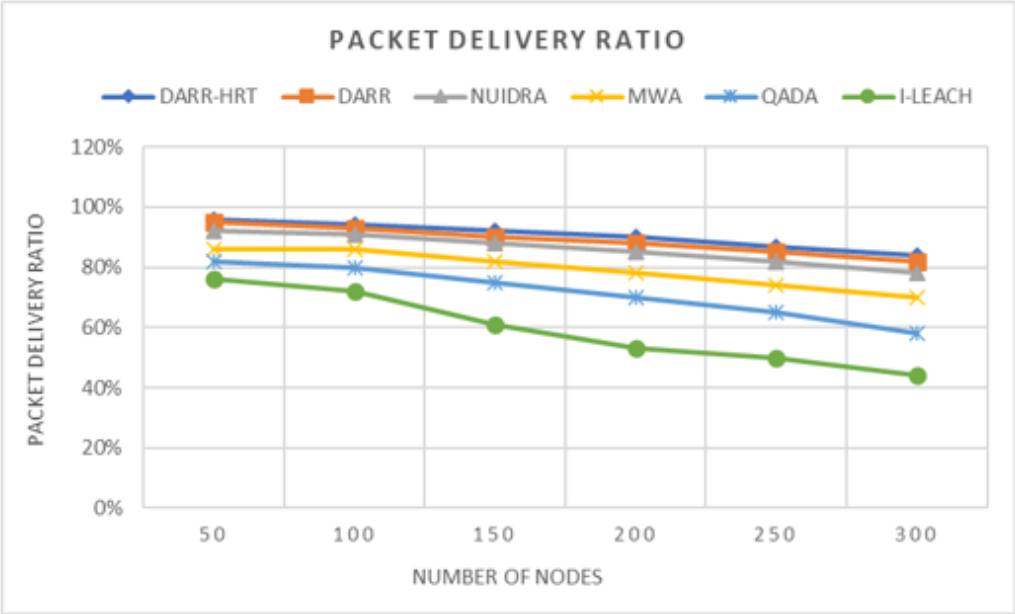

Figure 4.13: Effect of varying Node-density on Packet Delivery Ratio

PDR generally decreases as the number of nodes rises as a result of congestion, collisions, and energy consumption. DARR, DARR-HRT and NUIDRA protocols alleviate this deterioration more effectively than other protocols. MWA is a superior option for high level of reliability in dense IoT networks, i.e. network of more than 200 nodes).

Effect of varying Node-density on Throughput

As depicted in Figure 4.14, by eliminating duplicate transmissions, DARR increases throughput upto 13% as compared to NUIDRA alone. With the addition of high redundancy removal technique, DARR-HRT achieves a throughput that is 20-24% higher than NUIDRA. Compared to NUIDRA, throughput in MWA and QADA is lower by 31-50% and 21-24% respectively. Data must make several hops through cluster heads in MWA, which increases congestion and retransmissions. In dense deployments, QADA's hybrid cluster-tree routing performs better than I-LEACH. I-LEACH has a throughput that is 36-43% lower than NUIDRA because of static TDMA scheduling and energy gaps. DARR-HRT combines adaptive routing and redundancy elimination to maintain the best performance (1,042 kbps at 300 nodes). At 300 nodes, I-LEACH decreases to 480 kbps because of collisions and ineffective clustering.

End-to-End Delay

DARR shows lower end-to-end delay as observed in Figure 4.15. It reduces transmissions and collisions by combining DARR's redundancy reduction with NUIDRA's UI-based routing. Since there are less retransmissions at 600 seconds, the end-to-end delay is 1,800 ms as opposed to 2,100 ms with NUIDRA.

DARR-HRT have lowest end-to-end delay. It improves multi-hop routing speed by adding high redundancy removal temporal (HRT) mechanism to reduce packet size by up to 60%. For instance, for 600 sec, the delay is 2,300 ms as opposed to 2,900 ms for NUIDRA. QADA provide high end-to-end delay compared to NUIDRA. Control overhead, tree creation and multi-hop CH routing add a lot of delay to QADA. For instance, at 300 sec, the latency is 2,900 ms as opposed to 1,900 ms for NUIDRA. I-LEACH provides highest end-to-end delay. In I-LEACH, Queuing delays and collisions are caused by frequent CH rotations and static TDMA scheduling. At 600 sec, the delay is 5,350 ms as opposed to 2,900 ms for NUIDRA. While MWA and QADRA are more appropriate for adaptive yet latency-tolerant networks, DARR-HRT is the best option in largely scalable networks.

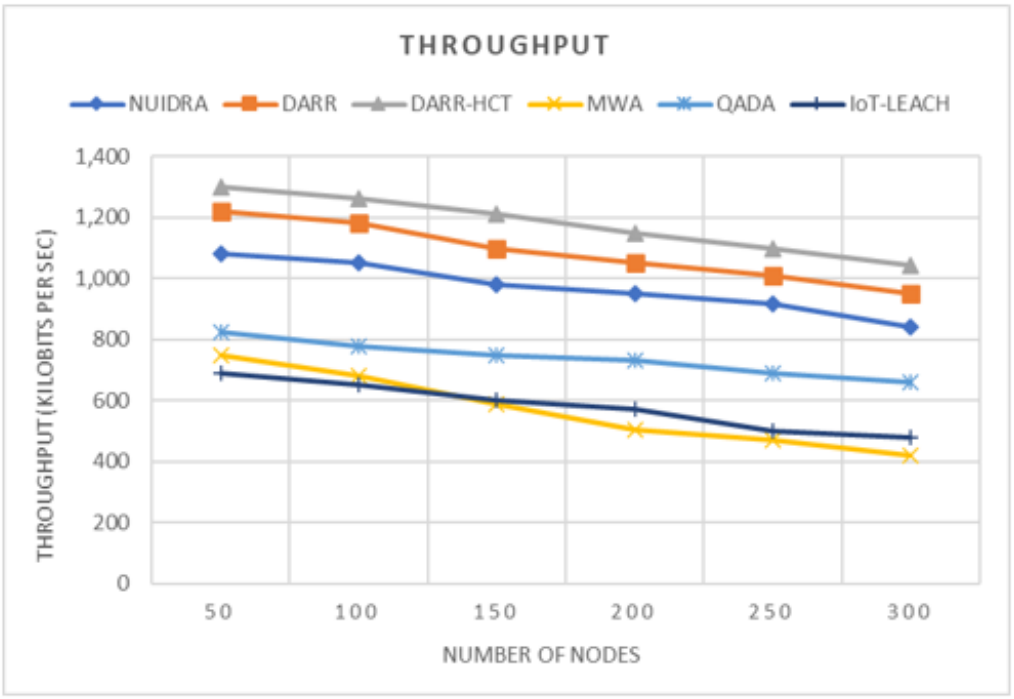

Figure 4.14: Effect of varying Node-density on Throughput

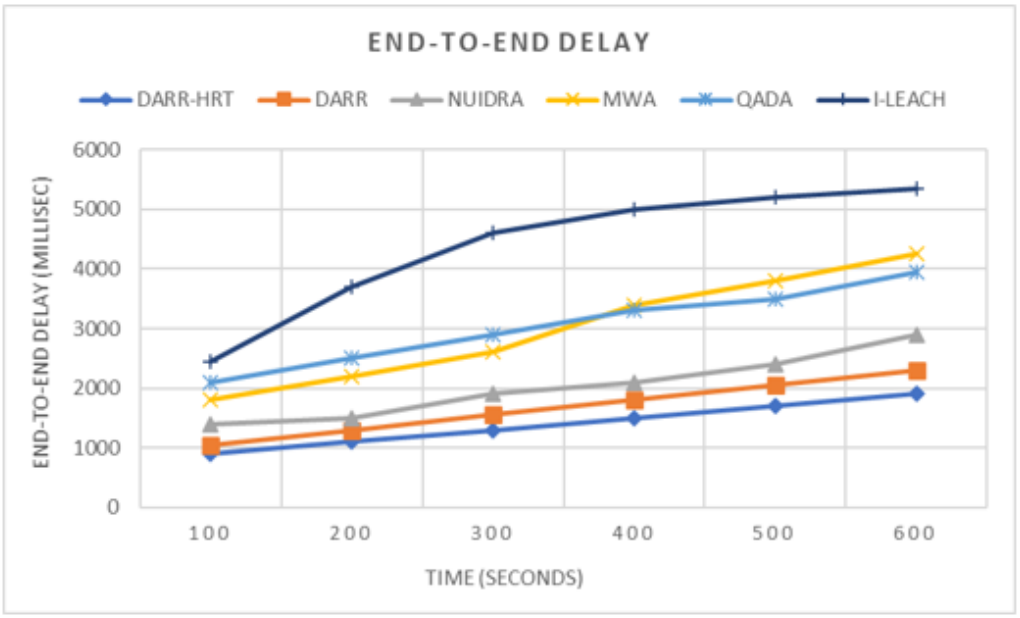

Figure 4.15: Analysis of End-to-End Delay over Time

Residual energy

The graph of Number of rounds verses Residual energy is shown in Figure 4.16. In DARR, sleep scheduling cuts transmissions by reducing the number of active nodes from 24 to 40%. In this case, because there are less redundant transmissions, the residual energy is 12-52% more than that for NUIDRA. In DARR-HRT for optimal energy savings, temporal compression (HRT) minimises energy use by reducing packet size by upto 60%. There is an increase in residual energy by 15-67% than NUIDRA. In MWA there is an increase in residual energy by 10-38% than NUIDRA as redundant data removal at node and CH provide less number of transmissions. While UI metrics in NUIDRA balance load, real-time computations consume energy. In QADA, tree construction increases overhead. Efficiency is decreased due to hybrid routing, which adds transmission of more control packets thereby decreasing the residual energy of the nodes. QADA shows from 5-33% reduction in residual energy

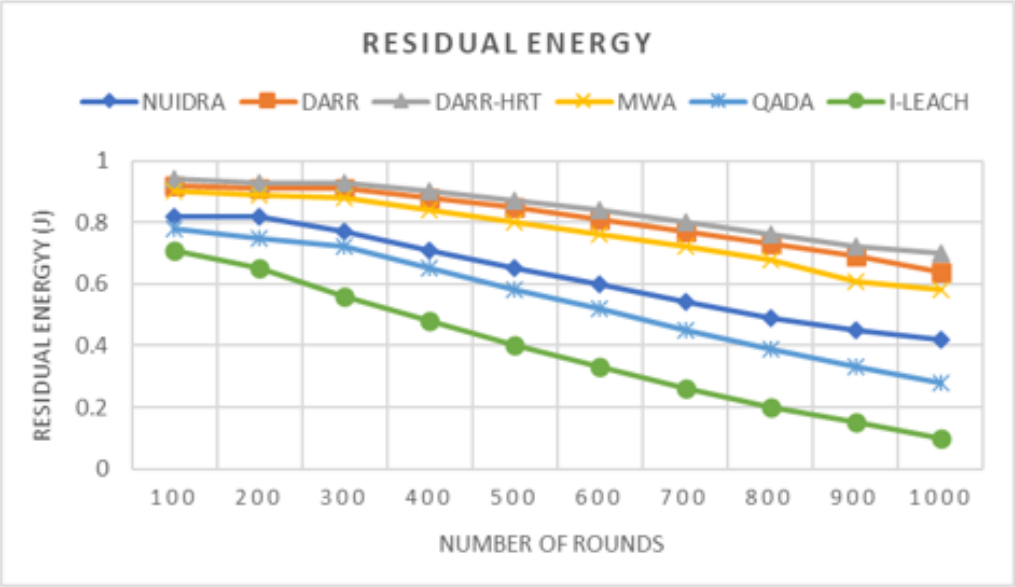

Figure 4.16: Analysis of Residual Energy

than NUIDRA . IoT-LEACH involves Static Clustering. CH rotations and fixed TDMA schedules quickly deplete energy. It shows 13-76% less residual energy than NUIDRA.

4.3 Summary of DARR protocol

The proposed DARR protocol [26] incorporates periodic monitoring of the Node-Alive Index to optimize network energy consumption by systematically managing redundant nodes. Additionally, it utilizes temporal correlation to aggregate data at each source node. DARR further enhances efficiency by eliminating redundant data from consecutive time slots at the source node level, resulting in fewer data transmissions. Moreover, the Hierarchical Redundancy Temporal (HRT) mechanism effectively reduces redundant data, resulting in minimal aggregated data transmissions and significantly extending network longevity.

To transmit data from the source nodes to the aggregator nodes and finally to the sink node, the DARR protocol uses the shortest and most reliable path computed in our earlier proposed NUIDRA protocol [23]. The data is aggregated at the aggregator nodes using spatial correlation [12] at the network level, which reduces the energy consumption of the network.

4.4 Application Scenario

The DARR (Data Aggregation with Removal of Redundancy) protocol [26] is a very adaptive and energy-efficient solution designed for application in wireless sensor networks (WSNs) where the sensor nodes are deployed randomly or densely and generate temporally or spatially redundant data. Its inherent ability to aggregate and filter data in an intelligent way makes it very suitable for a wide variety of real-time environmental monitoring agriculture monitoring, smart city deployments such as traffic or noise monitoring and smart infrastructure applications.

Flood management

The state of Uttarakhand suffered a mega disaster in the year 2013. It was one of the biggest disasters in living memory, causing widespread damage and destruction of lives and property. The impact of disaster was more pronounced in Kedarnath shrine valley. Heavy rain fall and cloud burst caused the collapse of Chaurabari Lake, resulting in flash flood in Kedarnath shrine temple and nearby areas of Mandakini river valley. As per the report of state government total of 169 people died and 4021 people

were reported missing (presumed to be dead) [31]. This disaster is also known as glacier lake outburst floods (GLOFs) and Himalayan Tsunami [31], [32]. WSNs can be used for flood management using water level monitoring. The wireless sensor network can be employed in any location in Uttarakhand having Glacier Lake and valley. Sensor nodes can be deployed at the side of area of interest (Glacier Lake). The sensor can be pressure sensor, piezoelectric sensor, vibration sensor and a water level detector, which will continuously measure the parameter and send this information to the local sink. The DARR protocol can be effectively implemented wherein sensor nodes and the sink node are connected to each other through wireless link. The water levels are monitored by the sensor nodes every hour and stored in its memory. When the readings for next hour are sensed then they are compared with earlier reading. Likewise the readings are compared on hourly basis and redundant data is removed at the aggregator node. The frequency of storing the sensed data from hourly basis can be reduced to 30 minutes or 15 minutes depending on the climatic conditions. If the difference of the sensed data in the previous interval and current interval is greater than the threshold value then the aggregator node will send it to the sink node. Sink node will transmit collected data from various sensor node to base station. The communication path between sink and base station can be normal wireless link or WiMAX [33]. Base station will transmit this information to gateway, which will be further forwarded to local server and backbone network. Backbone network will forward the information to nodal center where decisions will be taken. Local server and data base will be useful in case of backbone network failure and it will be monitored by local administration. In case of any abnormality in the level and pressure of water in the lake, precautionary measures can be taken prior to disaster and damage can be minimized.

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

The proposed research work prioritizes the development of energy-efficient, reliable protocols for distributed networks, specifically addressing data aggregation, routing, and redundant data elimination. The key results for the NUIDRA, DARR and DARR-HRT protocol are as follows.

- **Load Balancing**
  The NUIDRA protocol identifies highly utilized nodes, and redirects traffic to underutilized nodes, optimizing load distribution. It further eliminates redundant data at aggregation points before transmitting to the sink node. This dual strategy balances network-wide energy usage, prevents early node failure, and extends operational longevity.
  Result : Enhanced Network Lifetime
  In a 600-second simulation with 100 nodes (512-byte packets), NUIDRA sustains 26% active nodes, outperforming I-LEACH (10%) and QADA (18%).

- **Energy-Hole Mitigation**
  DARR resolves the energy-hole problem by waking up the slept redundant nodes and diverting traffic from the active heavily utilized nodes. If redundant nodes are not there then NUIDRA reduce the traffic from those heavily utilized active nodes by reducing the load balancing factor. The resulting balanced energy expenditure extends network lifespan and high residual energy.

  Result : Elevated Residual energy
  After 1,000 data transmission rounds with 300 nodes, NUIDRA retains 33% and 76% more residual energy than QADA and I-LEACH, respectively.

- **Optimal Path Routing**
  In NUIDRA, for finding shortest route, each node evaluates available routes to the sink and selects the path with the minimum hop count, while also considering real-time bandwidth usage.

  Result : Enhanced Packet Delivery Ratio and Throughput
  NUIDRA achieves superior packet delivery ratio (PDR) and throughput by minimizing collisions through its optimized routing strategy. This strategy dynamically assigns the shortest transmission path to each node based on hop count to the sink and real-time utilization metrics. In a 300-node deployment, NUIDRA attains a 78% PDR, outperforming MWA (70%), QADA (58%), and I-LEACH (44%). Additionally, NUIDRA delivers 50%, 21%, and 43% higher throughput than these protocols respectively under identical conditions.

- **Node Scheduling**
  The DARR protocol employs a spatial distance-based technique to identify redundant nodes

within the network. Nodes deemed redundant are transitioned into a Sleep state, significantly reducing unnecessary data transmissions and, in turn, lowering routing overhead. Unlike conventional protocols, which often leave redundant nodes active, this approach efficiently minimizes energy consumption and extends network operational lifespan.

Result : Extended Network Lifetime
The DARR protocol extends network lifetime by 31% compared to STCDRR and by 58% relative to DA-AFM when evaluated over 1,000 rounds of data transmission. This substantial improvement highlights DARRs effectiveness in sustaining network operation and conserving energy in long-term deployments.

- **Redundancy Elimination**
  The DARR protocol performs data aggregation at the source node level. By leveraging temporal correlation between sensed data, the protocol eliminates redundant information before transmission. This node-level aggregation drastically reduces the volume of data transmitted across the network, thereby conserving energy and optimizing resource utilization.

  Result: Minimization of Data Traffic
  For a similarity threshold value of 0.1 and 50 data measurements, the DARR, STCDRR, and DA-AFM protocols achieve dataset reductions of 48%, 43%, and 42%, respectively. Higher similarity thresholds correlate with increased dataset compression across all protocols.

  Result: Network Energy Optimization
  Scalability testing revealed DARR-HRT's superior energy efficiency. For a 500-node network transmitting 512-byte packets, DARR-HRT demonstrates substantial energy efficiency improvements: it consumes 88% less energy than DARR, and outperforms STCDRR and DA-AFM by 135% and 137% respectively across equivalent data rounds.

A comprehensive evaluation was conducted on key control parameters, including the effect of the redundancy threshold on identifying redundant nodes, varying packet size on network energy, the influence of the node-alive threshold on overall network performance, and the roles of spatial correlation and temporal similarity thresholds in shaping data measurement, aggregation, and network efficiency.

Both DARR and NUIDRA protocols demonstrated optimal energy management, minimized end-to-end delay, and improved reliability, establishing them as strong choices for contemporary WSN deployments that demand energy efficiency and extended operational lifetime.

5.2 Future Scope

As research continues to address issues with energy economy, scalability, and real-time processing, the future of data aggregation in Wireless Sensor Networks (WSNs) is expected to see considerable advancements. More intelligent and flexible aggregation techniques should be made possible by emerging trends like the integration of edge computing, machine learning, and cloud-based analytics. This will limit the redundant transmissions and increase network lifetime.

- **Dynamic Adjustment of Redundancy Threshold**
  Limitation: The redundancy threshold used to identify redundant nodes in DARR is configurable but remains static once set. This may not be optimal for applications with varying data generation rates or environmental conditions.
  Future Work: Implement adaptive thresholding mechanisms that dynamically adjust the redundancy threshold based on network traffic, energy levels, or application-specific requirements.

- **Active Node Sleep Interval Control**
  Limitation: If the interval between consecutive data sensing events is relatively long, nodes remain in an active state throughout this duration, leading to continuous energy consumption

and accelerated battery depletion.
Future Work: To address this, when the interval between sensing events is substantial, nodes can be transitioned into a SLEEP state during idle periods. Implementing such adaptive sleep scheduling would help conserve energy in resource-constrained nodes and extend overall network lifetime.

- **Scalability to Larger Networks**
Limitation: The current implementation of DARR and NUIDRA is tested on medium-scale networks. As the network size increases, the complexity of identifying redundant nodes and computing optimal routes may grow, potentially impacting performance.
Future Work: Explore advanced optimization techniques, such as machine learning-based clustering or distributed processing, to scale the protocols effectively for larger networks with thousands of nodes.

- **Interoperability with IoT Systems**
Limitation: The protocols are tailored for standalone WSNs and do not account for integration with broader Internet of Things (IoT) ecosystems.
Future Work: Enhance the protocols to support interoperability with IoT platforms by incorporating standardized communication protocols and enabling cloud-based data analytics.

In order to ensure reliable, scalable, and efficient wireless sensor network deployments, novel data aggregation approaches will be essential as WSNs become more and more integrated into smart cities, precision agriculture, and industrial automation. Furthermore, for applications in sensitive situations, the creation of secure and privacy-preserving methods will be essential.

Publications

- **Publication - Journals**

- A. Bomnale, and A. More, Node utilization index-based data routing and aggregation protocol for energy-efficient wireless sensor networks, The Journal of Supercomputing, Springer, vol. 80, no. 7, pp. 9220-9252, 2023.
  (Web of Science, Scopus Indexed, Impact Factor - 3.3)

  https://doi.org/10.1007/s11227-023-05800-4

- A. Bomnale, and A. More, Node-Alive index driven redundancy elimination for energy-efficient wireless sensor networks, EAI Endorsed Transactions on Scalable Information Systems, European Alliance Inc. Vol. 11 No.6, pp.1 -20, EAI, 2024.
  (Web of Science, Scopus Indexed Impact Factor 1.3)

  https://publications.eai.eu/index.php/sis/article/view/7397

- A. Bomnale, and A. More, A survey of data aggregation and routing protocols for energy-efficient wireless sensor networks", EAI Endorsed Transactions on Scalable Information Systems, European Alliance for Innovation, Vol. 12 No. 2, pp. 1-23, EAI, Apr 2025
  (Web of Science indexed, Impact Factor : 1.3)

  https://publications.eai.eu/index.php/sis/article/view/6924/3551

- A. Bomnale, and A. More, Environmental Monitoring Applications using Energy-efficient Data Aggregation and Routing protocol in Wireless Sensor Networks, International Journal of Wireless Information Network, Springer. Submitted on 16 Jun 2025
  Under Review

Bibliography

[1] V. C. M. W. X.-W. S. K. Biswas, K; Muthukkumarasamy, "A multipath routing protocol for secure energy efficient communication in wireless sensor," *Computer Networks, Elsevier*, vol. 232, p. 109842, 2023.

[2] S. R. Jan, R. Khan, F. Khan, M. A. Jan, M. D. Alshehri, V. Balasubramaniam, and P. S. Sehdev, "Marginal and average weight-enabled data aggregation mechanism for the resource-constrained networks," *Computer Communications, Elsevier*, vol. 174, pp. 101–108, 2021.

[3] S. Kurian and L. Ramasamy, "Novel aodv based service discovery protocol for manets," *Wireless Networks, Springer*, vol. 27, no. 4, pp. 2497–2508, 2021.

[4] F. Fanian and M. K. Rafsanjani, "Cluster-based routing protocols in wireless sensor networks: A survey based on methodology," *Journal of Network and Computer Applications*, vol. 142, pp. 111–142, 2019.

[5] J. Zhang and Z. Sun, "Assessing multi-hop performance of reactive routing protocols in wireless sensor networks," in *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, pp. 444–449, IEEE, 2016.

[6] P. Marappan and P. Rodrigues, "An energy efficient routing protocol for correlated data using cl-leach in wsn," *Wireless Networks, Springer*, vol. 22, no. 4, pp. 1415–1423, 2016.

[7] X. Z. Wang, "The comparison of three algorithms in shortest path issue," in *Journal of Physics: Conference Series*, vol. 1087, pp. 011–022, IOP Publishing, 2018.

[8] D. Kothandaraman and C. Chellappan, "Energy efficient node rank-based routing algorithm in mobile ad-hoc networks," *International Journal of Computer Networks and Communications*, vol. 11, no. 1, pp. 45–61, 2019.

[9] T. Sawant and S. Sirsikar, "A comparative study of various routing technique for wireless sensor network with sink and node mobility," *Intelligent Communication and Computational Technologies, Springer*, pp. 227–236, 2018.

[10] M. R. Ghori, T.-C. Wan, G. C. Sodhy, and A. Rizwan, "Optimization of the aodv-based packet forwarding mechanism for ble mesh networks," *Electronics*, vol. 10, no. 18, p. 2274, 2021.

[11] T. Lee, D. S. Kim, H. Choo, and M. Kim, "A delay-aware scheduling for data aggregation in duty-cycled wireless sensor networks," in *2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks*, pp. 254–261, IEEE, 2013.

[12] R. Maivizhi and P. Yogesh, "Spatial correlation based data redundancy elimination for data aggregation in wireless sensor networks," in *2020 International Conference on Innovative Trends in Information Technology (ICITIIT)*, pp. 1–5, IEEE, 2020.

[13] L. Dash, B. K. Pattanayak, S. K. Mishra, K. S. Sahoo, N. Z. Jhanjhi, M. Baz, and M. Masud, "A data aggregation approach exploiting spatial and temporal correlation among sensor data in wireless sensor networks," *Electronics*, vol. 11, no. 7, p. 989, 2022.

[14] K. Sekar, K. Suganya Devi, and P. Srinivasan, "Energy efficient data gathering using spatio-temporal compressive sensing for wsns," *Wireless Personal Communications, Springer*, vol. 117, no. 2, pp. 1279–1295, 2021.

[15] B. Yamini, P. G, K. D, J. M, J. G, and U. G S, "Theoretical study and analysis of advanced wireless sensor network techniques in internet of things (iot)," *Measurement: Sensors*, vol. 33, p. 101098, 2024.

[16] P. Chugh, M. Gupta, S. Indu, G. Chaudhary, M. Khari, and V. Shanmuganathan, "Advanced energy efficient pegasis based routing protocol for iot applications," *Microprocessor,s Microsystem., Elsevier*, vol. 103, Mar. 2024.

[17] J. Agarkhed, V. Kadrolli, and S. R. Patil, "Efficient bandwidth-aware routing protocol in wireless sensor networks (ebarp)," *International Journal of Information Technology*, pp. 1–13, 2022.

[18] H. Gizachew Yirga, G. Desalegn Taye, and H. Mulugeta Melaku, "An optimized and energy-efficient ad-hoc on-demand distance vector routing protocol based on dynamic forwarding probability (aodvi)," *Journal of Computer Networks and Communications, Hindawi*, vol. 2022, no. 1, p. 5750767, 2022.

[19] W.-K. Yun and S.-J. Yoo, "Q-learning-based data-aggregation-aware energy-efficient routing protocol for wireless sensor networks," *IEEE Access*, vol. 9, pp. 10737–10750, 2021.

[20] M. E. Haque and U. Baroudi, "Dynamic energy efficient routing protocol in wireless sensor networks," *Wireless Networks, Springer*, vol. 26, p. 37153733, July 2020.

[21] P. Joshi, A. S. Raghuvanshi, and S. Kumar, "An intelligent delay efficient data aggregation scheduling for distributed sensor networks," *Microprocessor, Microsystem. Elsevier*, vol. 93, sep 2022.

[22] T.-D. Nguyen, D.-T. Le, V.-V. Vo, M. Kim, and H. Choo, "Fast sensory data aggregation in iot networks: Collision-resistant dynamic approach," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 766–777, 2020.

[23] A. Bomnale and A. More, "Node utilization index-based data routing and aggregation protocol for energy-efficient wireless sensor networks," *The Journal of Supercomputing, Springer*, vol. 79, pp. 1–33, 2023.

[24] H. Rahman, N. Ahmed, and M. I. Hussain, "A qos-aware hybrid data aggregation scheme for internet of things," *Annals of Telecommunications, Springer*, vol. 73, pp. 475–486, 2018.

[25] T. M. Behera, U. C. Samal, and S. K. Mohapatra, "Energy-efficient modified leach protocol for iot application," *IET Wireless Sensor Systems*, vol. 8, no. 5, pp. 223–228, 2018.

[26] A. Bomnale and A. More, "Node-alive index driven redundancy elimination for energy-efficient wireless sensor networks," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 12, no. 1, 2025.

[27] K. Haseeb, N. Islam, T. Saba, A. Rehman, and Z. Mehmood, "Lsdar: A light-weight structure based data aggregation routing protocol with secure internet of things integrated next-generation sensor networks," *Sustainable Cities and Society*, vol. 54, p. 101995, 2020.

[28] T. S. Rappaport, *Wireless Communications: Principles and Practice.* Upper Saddle River, NJ: Prentice Hall, 2nd ed., 2002.

[29] "The network simulator ns-2." http://www.isi.edu/nsnam/ns/, last used: 2019-24.

[30] K. Jain and A. Kumar, "Energy-efficient data-aggregation technique for correlated spatial and temporal data in cluster-based sensor networks," *International Journal of Business Data Communications and Networking (IJBDCN)*, vol. 16, no. 2, pp. 53–68, 2020.

[31] National Institute of Disaster Management, Ministry of Home Affairs, Government of India, "Uttarakhand Disaster Report." http://nidm.gov.in/PDF/pubs/ukd-p1.pdf, 2013. Accessed: 2025-06-15.

[32] National Institute of Disaster Management, Ministry of Home Affairs, Government of India, "India Disaster Report." http://nidm.gov.in/PDF/IEC/leaflet-dos.pdf, 2013. Accessed: 2025-06-15.

[33] G. Krishna, A. Raii, and S. Saxena, "Land slide monitoring and hazard mapping in uttarakhand using reinforcement learning and neural network," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, August 2013.